THE FORWARD RESERVE WAREHOUSE SIZING AND DIMENSIONING PROBLEM

A Dissertation
Presented to
The Academic Faculty

by

Jinxiang Gu

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Industrial and Systems Engineering
Georgia Institute of Technology
December 2005

THE FORWARD RESERVE WAREHOUSE SIZING AND DIMENSIONING PROBLEM

Approved by:

Professor Leon F. McGinnis
School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor Marc Goetschalckx
School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor Gunter P. Sharp
School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor Joel Sokol
School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor Christiaan Paredis
School of Mechanical Engineering
Georgia Institute of Technology

Date Approved: 1 September 2005

ACKNOWLEDGEMENTS

I am very grateful to my advisors, Dr. Leon McGinnis and Dr. Marc Goetschalckx, for their numerous supports for my study in the School of Industrial and Systems Engineering at Georgia Institute of Technology. The PhD study is a long and sometimes stressful journey and would not be successful if without their guidance, encouragement, and patience. I learned a lot from them both academically and personally, which will become an invaluable asset in my future career life. I am also indebted to the other committee members, Dr. Gunter Sharp, Dr. Joel Sokel, and Dr. Christiaan Paredis, for their insightful comments and suggestions that helped improving this research.

There are too many friends to name who have helped me during the past several years. Special thanks go to Dr. Doug Bodner, who spent a lot of time in coordinating the research activities and providing technical supports for students like me in the Virtual Factory Laboratory.

Finally, I would like to thank my parents, Denghong Gu and Xiaomei He, and my wife, Na An. They have always been a tremendous source of encouragement in my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This research addresses sizing and dimensioning of a forward-reserve warehouse, a strategic design problem that has important implications for warehouse life cycle costs including construction, inventory holding and replenishment, and material handling. Large mixed integer nonlinear models are developed that capture the complex tradeoffs among the different costs in order to achieve a global optimal design satisfying throughput requirements. We first consider the situation where the forward area includes all SKUs so that order picking is performed only in the forward area. In this case, the problem can be decomposed and the resulting sub-problem is convex and can be solved very efficiently based on the Karush-Kuhn-Tucker (KKT) conditions. This property enables the use of a Generalized Benders Decomposition (GBD) method to solve the sizing and dimensioning problem exactly.

We then extend the problem to more general situations where the forward area can contain a subset of SKUs. This requires integrating the sizing and dimensioning decisions and the decision to assign SKUs to the forward area based on their flow characteristics (i.e., the forward reserve allocation). A similar decomposition strategy can be employed, but the sub-problem (incorporating the forward reserve allocation) is no longer convex. A bi-level hierarchical heuristic approach is proposed that integrates a pattern search method for the master problem and optimal and heuristic algorithms for the sub-problems. Numerical results demonstrate that the proposed heuristic approach is very efficient in generating near optimal solutions.

A detailed discussion of the forward reserve allocation problem is also provided since it appears in the sizing and dimensioning problem as a sub-problem. The forward reserve allocation problem is NP-complete itself and is solved heuristically in the previous literature. An alternative branch-and-bound algorithm based on outer approximation is developed that can quickly find the optimal solution for realistically sized problems. Extensive numerical experiments based on real warehouse data are conducted to compare the heuristic and optimal solutions. The results show that, although the optimality gap might be big in some small examples, for realistic warehouses the heuristic solution is always very close to the optimal solution in terms of both the objective value and the forward assignment.

CHAPTER 1

INTRODUCTION

Warehouses are an essential component of any supply chain. Their major roles include: buffering the material flow along the supply chain to accommodate variability caused by factors such as product seasonality and/or batching in production and transportation; consolidation of products from various suppliers for combined delivery to customers; and value-added-processing such as kitting, pricing, labeling, and product customization.

Market competition requires continuous improvement in the design and operation of production-distribution networks, which in turn requires higher performance from warehouses. The adoption of new management philosophies such as Just-In-Time (JIT) or lean production also brings new challenges for warehouse systems, including tighter inventory control, shorter response time, and a greater product variety. On the other hand, the widespread implementation of new information technologies (IT), such as bar coding, radio frequency communications (RF), and warehouse management systems (WMS), provides new opportunities to improve warehouse operations. These opportunities include, but are not limited to: real-time control of warehouse operation, easy communication with the other parts of the supply chain, and high levels of automation.

Warehouse design and operation have attracted a lot of research attention in the last two decades. Nevertheless they remain complex tasks with few useful decision support tools. The difficulties include: the large amount of information to be processed; the large number of possible alternatives; various and often conflicting objectives; and uncertainty inherent in the material flow into, through, and out of the warehouse.

Furthermore, as the literature survey in chapters 2 and 3 shows, most research efforts have been devoted to warehouse operation problems instead of design problems. This is not surprising since design problems are much more difficult to treat analytically. A major difficulty is that design decisions are tightly coupled and good designs require considering them in an integrated way. However, integrated design models are more difficult to develop and analyze.

This dissertation addresses a strategic design problem, i.e., the sizing and dimensioning problem for warehouses with the forward reserve configuration. The forward-reserve configuration is a popular warehouse layout strategy that allows both efficient order picking and efficient storage. It divides the warehouse into a forward area and a reserve area. The forward area is mainly used for order picking and is characterized by: (1) expensive storage and picking equipment that facilitates convenient item selection and retrieval; (2) low storage density in terms of the net storage volume per unit storage area; (3) high picking efficiency in terms of the average travel and item retrieval time per pick. The reserve area is mainly used for bulk storage. It is also used for replenishing the forward area and for picking Stock Keeping Units (SKUs) that are not assigned to the forward area. Compared with the forward area, it has the following features: (1) inexpensive storage methods such as block stacking and pallet racks; (2) high storage density; and (3) low picking efficiency. The benefit of the forward-reserve configuration lies in the fact that it dedicates different storage areas to the two different and usually conflicting warehouse functions, i.e., order picking and storage, so that their relative advantage can be best utilized with the minimum interference between them. However, this benefit cannot be fully realized without careful storage space planning to balance the tradeoffs among the costs of equipment, inventory, and material handling (e.g., order picking, internal replenishment, put away).

The sizing and dimensioning problem determines the size and dimension of a warehouse, the space allocation between the forward and reserve areas, and the space allocation for SKUs within each area. It is a strategic decision that has important implications for warehouse life cycle costs, which include construction cost, inventory holding and replenishment cost, and material handling cost. This research is the first that develops integrated models to balance the complex tradeoffs between the different cost elements in order to achieve a global optimal design. Two scenarios are modeled: (1) the forward area contains all SKUs so that orders are picked only from the forward area; (2) the forward area contains a selected subset of SKUs, and customer requests can be picked from either area depending on the forward assignment. The sizing and dimensioning problem for both scenarios are modeled as large mixed integer nonlinear optimization problems. Their structures are explored in order to develop efficient and effective solution algorithms. For the first scenario, the problem is solved with a generalized Benders decomposition method that can find the guaranteed optimal solution. The algorithm for the second scenario is a hierarchical heuristic method that integrates pattern search for solving the master problem together with a bisection search method and a knapsack-based heuristic for solving the sub-problems. Numerical results will be provided with regards to the performance of the proposed algorithms as well as the quality of the resulting solutions.

The organization of the dissertation is as follows. Chapters 2 and 3 give a comprehensive literature survey on warehouse operation and design problems respectively. The scope of this survey is not limited to the specific problem we studied, but covers most of the important topics in the warehouse literature. Therefore, they can be regarded as results independent from the other chapters. Chapter 4 presents the warehouse sizing and dimensioning problem for scenario 1 as well as the GBD based global optimal algorithm. Chapter 5 discusses the forward reserve allocation problem, which is one of the sub-problems in the sizing and dimensioning problem for scenario 2. Chapter 6 then generalizes the model developed in chapter 4 to include the additional decision of forward assignment and provides a hierarchical heuristic solution method to solve the generalized model. Finally, research results are summarized and future directions are given in chapter 7.

CHAPTER 2

RESEARCH ON WAREHOUSE OPERATION

2.1 Introduction

A number of warehouse operation decision support models have been proposed in the literature, but there remains considerable difficulty in applying these models to guide warehouse operations. This chapter presents a comprehensive review of the state-of-the-art in research on warehouse operation planning. The objective is to classify and summarize the prior research results, and to identify the research opportunities for the future. The intended outcome is both a guide to practitioners on the analytical methodologies and tools available to support better warehouse operation planning, and a roadmap for academic researchers to future research opportunities.

We first present a unifying framework to classify the research on different but related warehouse planning problems. Within this framework, historical progress and major results are summarized with an emphasis on how the research on these problems evolved and the relationships between various problems. Future research directions are identified and discussed.

2.2 Framework

The basic requirements in warehouse operations are to receive goods from suppliers, store the goods, receive orders from customers, retrieve goods and assemble them for shipment, and ship the completed orders to customers. There are many issues involved in designing and operating a warehouse to meet these requirements. Resources, such as space, labor, and equipment, need to be allocated among the different warehouse functions, and each function needs to be carefully implemented, operated, and coordinated in order to achieve system requirements in terms of capacity, throughput, and service at the minimum resource cost.

A scheme to classify warehouse design and operation planning problems and the corresponding literature is shown in Figure 2.1 (the numbers in parentheses represent the number of papers reviewed for each operation planning problem) and a more detailed description of each problem category identified is given in Table 2.1. This chapter will focus on the operation planning problems, while warehouse design and performance evaluation are discussed in the next chapter.

Storage is concerned with the organization of goods held in the warehouse in order to achieve high space utilization and facilitate efficient material handling. Goods in storage can be organized in department based on physical characteristics of the goods (e.g., pallet storage vs. case storage), management considerations (e.g., a dedicated storage area for a specific customer), or material handling considerations (e.g., a forward area for fast picking). Within departments, goods may be further organized into pick zones. A pick zone is a set of storage locations that are often arranged in close physical proximity. A department may be divided into zones because of storage requirements, for example, when different block-stacking patterns are used for pallet storage. A department may also be divided into zones for organizing order picking activities. A particular pick zone holds a limited subset of the SKUs, and pickers are dedicated to their zone to pick the required items. Because of the limited physical size of the zone, the picker achieves a

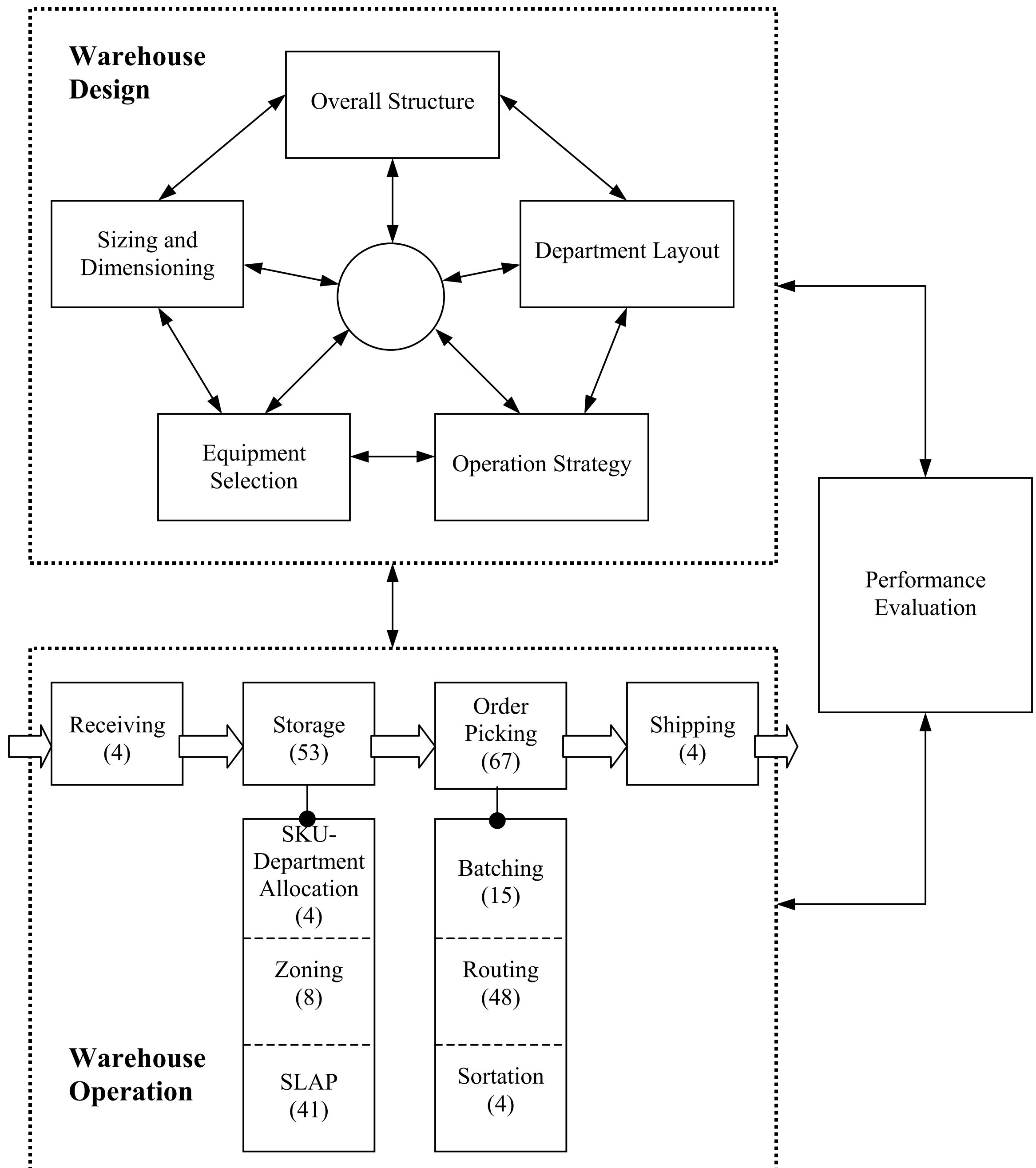


Figure 2.1 Framework for warehouse design and operation problems

Table 2.1 Description of warehouse design and operation problems

Design and Operation Problems			Decisions
Warehouse Design	Overall Structure		• Material flow • Department identification • Relative location of departments
	Sizing and Dimensioning		• Size of the warehouse • Size and dimension of departments
	Department Layout		• Pallet block-stacking pattern (for pallet storage) • Aisle orientation • Number, length, and width of aisles • Door locations
	Equipment Selection		• Level of automation • Storage equipment selection • Material handling equipment selection (order picking, sorting)
	Operation Strategy		• Storage rule selection • Order picking method selection
Warehouse Operation	Receiving & Shipping		• Truck-dock assignment • Order-truck assignment • Truck dispatch schedule
	Storage	SKU-Department Assignment	• Assignment of items to different warehouse departments • Space allocation
		Zoning	• Assignment of SKUs to zones • Assignment of pickers to zones
		Storage Location Assignment	• Storage location assignment • Specification of storage classes (for class-based storage)
	Order Picking	Batching	• Batch size • Order-batch assignment
		Routing & Sequencing	• Routing and sequencing of order picking tours • Dwell point selection (for AS/RS)
		Sortation	• Order-lane assignment

high ratio of SKU extracting time to traveling time between locations and an increased familiarity with the SKUs in the zone. Within a department /zone, goods are assigned to storage locations, and storage location assignment has significant impact on storage capacity, inventory tracking, and order picking. For example, dedicated storage (as defined in section 2.4.3) has low space utilization, but the warehouse is easier to manage since it has a permanent assignment of products to locations.

Order picking is generally recognized as the most expensive warehouse operation, because it tends to be either very labor intensive or very capital intensive (Frazelle (2001)). Managing the order picking process requires the organization of the orders to be picked and the material handling operations of the picking. In a given day or shift, a warehouse may have many orders to pick. These orders may be similar in a number of respects; for example, some orders are shipped using the same carrier, or transportation mode, or have the same pick due date and time. If there are similarities among subsets of orders that require them to be shipped together, then they also should be picked roughly during the same time period to avoid intermediate storage and staging. Thus, it is common practice to use wave picking, i.e., to release a fraction of the day's (shift's) orders, and to expect their order picking to be completed within a corresponding fraction of the day (shift).

In addition to wave picking, two other commonly used order-picking strategies are batch picking and zone picking. Batch picking involves the assignment of a group of orders to a picker to be picked simultaneously in one trip. In zone picking, the storage space is divided into picking zones and each zone has one or more assigned pickers who only pick in their assigned zone. Zone picking can be divided into sequential and parallel zone picking. Sequential zone picking is similar to a flow line, in which containers that can hold one or more orders are passed sequentially through the zones; the pickers in each zone pick the products within their zone, put them into the container, and then pass the container to the next zone. (Bartholdi and Eisenstein (1996) propose a Bucket Brigades order picking method that works similar to sequential zone picking, but does not require pickers to be restricted to zones). In parallel zone picking, an order is picked in each zone simultaneously. The picked items are sent to a downstream sortation system to be combined into orders.

The organization and planning of the order picking process must answer the following questions:

1. Will product be transported to the picker (part-to-picker) or will the picker travel to the storage location (picker-to-part)?
2. Will orders be picked in waves? If so how many waves of what duration?
3. Will the warehouse be divided into zones? If so, will zones be picked sequentially or concurrently?
4. Will orders be picked in batches or separately? If they are batched, will they be sorted while picking or after picking?

Depending on the operating principles selected, the order picking methods will be:

- Single order picking
- Batching with sort-while-pick
- Batching with sort-after-pick
- Sequential zoning with single order picking
- Sequential zoning with batching

- Concurrent zoning without batching
- Concurrent zoning with batching

Each of the above order picking methods can be decomposed into a set of basic steps. For example, using the batching with sort-after-pick method, orders arriving at the warehouse are first batched; each batch is routed and a picking list is generated; the batch is then assigned to an order picker who travels to the order picking area to pick the required items; the picked items are sent to a downstream sortation system to be sorted into orders; and finally the picked orders are packed and shipped to customers.

Therefore, several basic decisions need to be made for each picking method, which include pick wave sizing, batching, routing, and sorting. The planning of batching, routing, and sorting defines the basic decision making modules at the operational level for order picking and will be discussed in detail in Section 2.5. Research on pick wave sizing is very limited, and therefore will not be further discussed.

2.3 Receiving and shipping

Goods arrive to a warehouse in a carrier and are unloaded at the receiving docks. Later they are loaded in a carrier and leave the warehouse through the shipping docks. For cross docking warehouses, received goods are directly sent from the receiving docks to the shipping docks. For traditional warehouses that hold inventory received goods are put away into storage and later picked and shipped through shipping docks. In this case, the receiving and shipping operations are more complex to manage since they are coupled with the storage and order picking functions. For example, the scheduling of shipping trucks may depend on how orders are batched and assigned to picking waves and vice versa.

The basic decisions in receiving/shipping operations are:

- Assign inbound and outbound carriers to docks. This assignment determines the aggregate internal material flows.
- Schedule the service of carriers at each dock. Assuming a set of carriers is assigned to a dock, the problem is similar to a machine-scheduling problem, where the arriving carriers are the jobs to be scheduled.
- Allocate or dispatch material handling resources, such as labor and material handling equipment.

The objective of these decisions is to minimize the resources required to complete all shipping/receiving operations with acceptable levels of service. The criteria may vary according to the function of the warehouse, but typical criteria would include the total cycle time for the carriers, the load/unload time for the carriers, and the material handling cost associated with load/unload operations.

There are also a variety of constraints on dock operations, including:

- Layout, or the relative location and arrangement of docks and storage departments.
- Management policies, e.g., one customer per shipping dock.
- Finish time requirements for some customers or docks.
- Throughput requirements for all docks.

Furthermore, decision making is limited by the level of prior knowledge about incoming and outgoing shipments, for which the following scenarios can be distinguished:

- No knowledge, other than warehouse layout.
- Partial statistical knowledge of arriving and departing processes, such as the average level of material flow from an incoming carrier to an outgoing carrier.
- Perfect knowledge of the content of each arriving carrier and each departing carrier.

In the first scenario, not only do we have no basis for assigning carriers to docks, we also have no basis for assigning goods to storage locations. It is not clear in this case if any storage assignment rule is preferred. Public warehouse may operate under this set of conditions. The second scenario is most common in company-owned or dedicated distribution warehouses and is the basis for most of the decision models in the literature. The third scenario is becoming increasingly common through the application of advanced information technologies such as RFID, GPS, and advanced shipping notices (ASN).

The research on receiving and shipping has been focused on the carrier-to-dock assignment problem for cross-docking warehouses, assuming statistical knowledge of incoming and outgoing shipments. The cross-docking warehouse is operated as follows: inbound trucks arrive in the yard of the warehouse and proceed to the assigned receiving door (or strip door) for unloading; the unloaded goods are sorted according to their destinations, and then loaded onto outbound trucks at shipping doors (or stack doors) for delivery to customers. Often, each stack door is designated to a particular destination, and once established, the designations of stack doors generally do not change. The decisions for a cross-docking warehouse manager are then to designate the doors as either strip or stack doors, assign destinations to stack doors, and assign inbound trucks to strip doors in order to minimize the total operational cost.

Assuming the designations of doors as either strip or stack doors have already been made, Tsui and Chang (1990, 1992) formulate a bilinear model to assign inbound and outbound trucks to strip and stack doors respectively. Gue (1999) proposes a model to estimate the operational cost by optimally assigning inbound trucks to strip doors given the specification of doors as either strip or stack doors and the assignment of destinations to stack doors. Based on the cost model, he uses a local search procedure to find an efficient door layout. Bartholdi and Gue (2000) consider the cross-docking warehouse door layout problem with the objective of minimizing the total travel time and waiting time incurred due to congestion. They model the total travel time and waiting time for a fixed door layout using transportation and queuing models and then embed the cost model in a simulated annealing algorithm to find an efficient door layout.

In summary, very few formal models have been developed for the management of shipping and receiving operations. Most of the literature that is available in this area addresses shipping and receiving operations and truck-to-dock assignment strategies for cross-docking warehouses.

2.4 Storage

Storage is a major warehouse function. Three decisions have to be made regarding the storage of goods in a warehouse, i.e., how much inventory should be kept in the warehouse for an SKU; how frequently and at what time should the inventory for an SKU be replenished; and where should the SKU be stored in the warehouse and distributed and moved among the different storage areas. The first two problems are called the lot sizing and staggering problems respectively, which belong to the traditional inventory control area and are not further discussed here. Readers may refer to Gallego et al. (1996) and Hariga and Jackson (1996) for a detailed review. This section will focus on the storage assignment problem, which includes the decisions of assigning SKUs to various storage departments and scheduling of inventory moves between the departments, of assigning SKUs to different zones (zoning), and of the storage location assignment within a department/zone. The two major criteria in making these decisions are the storage efficiency, which corresponds to the holding capacity, and the access efficiency, which corresponds to the resources consumed by the insertion (store) and extraction (order picking) processes.

2.4.1 Assigning SKUs across departments

A SKU may be stored in more than one warehouse department. The specification of departments is a design decision. Once the departments are specified, one needs to determine which SKU should be stored in which department, in what quantity, and what are the corresponding inter-departmental moves for that SKU. In some cases, this decision is straightforward. For example, if a department is dedicated to a certain customer, then all SKUs for that customer are assigned to that department; or if a SKU will be stored and picked only in units of pallets, then it will be assigned only to the pallet storage area. In other cases, a SKU could be assigned to multiple departments. These departments usually differ in terms of their storage and material-handling capability.

Therefore, a careful decision needs to be made in order to balance the tradeoff between storage and material handling cost and capacities.

The forward-reserve problem belongs to this category and is a well-researched problem. It is a common practice in warehousing to create a separate, physically compact forward (or "fast pick") area for picking high-demand, fast-moving products. This reduces order picking costs but at the expense of requiring additional material handling to restock the forward area from a reserve area. Furthermore, the size of the forward area is limited. Therefore, one needs to determine which SKUs should be stored in the forward area and in what quantity in order to achieve the maximum savings in material handling.

Bozer (1985) first introduces the problem of splitting a pallet rack into an upper reserve area and a lower forward picking area. Hackman and Rosenblatt (1990) treat the problem of deciding which SKUs to assign to the forward area, and how to allocate space among the assigned SKUs, given the forward area has a fixed capacity. The objective is to minimize the total material handling costs of order picking and replenishing. They propose a knapsack-based heuristic to solve this problem and provide sufficient conditions for optimality of this heuristic. Frazelle et al. (1994) extend the problem and solution method of Hackman and Rosenblatt (1990) by treating the size of the forward area as a decision variable. The costs in their model include the equipment cost of the fast pick area (modeled as a linear function of its size), and the material handling cost for order-picking and replenishment.

The above models assume the replenishment of a SKU can be done in a single trip. van den Berg et al. (1998) consider the problem for unit-load replenishments, i.e., only one unit can be replenished per trip. Assuming the forward area can be replenished instantaneously there is no need to assign more than one unit to the forward area. They consider warehouses that have busy and idle periods, so it is possible to reduce the number of replenishments during busy periods by performing replenishments in the preceding idle periods. A knapsack-based heuristic is proposed to find the set of SKUs to put in the forward area that minimizes the expected total labor-time related to order-picking and replenishing during a busy period.

2.4.2 Assigning SKUs across zones (zoning)

The zoning problem is to specify different storage zones within a department and assign SKUs to the specified zones. It can be both a “hard” and a “soft” decision; it is a hard decision if it leads to zone-specific storage technology selection and physical arrangement, but it is a soft decision if it is simply an organization of similar storage locations. Thus, zoning decisions fall in between warehouse design decisions and warehouse operation decisions.

A department can have zones that use different storage modes. One example that has been extensively studied in the literature is the block-stacking problem for a pallet storage area. A fundamental decision for the block-stacking problem is to determine lane depths to balance the tradeoffs between space utilization and ease of storage/retrieval operations, considering the SKUs’ stackability limits, arriving lot sizes, and retrieval patterns. Using deep lane storage might increase space utilization because fewer aisles are needed, but on the other hand might also decrease space utilization due to the “honeycombing” effect that results in unused space in the lane that cannot be assigned to other items before the whole lane is totally depleted. Honeycombing effect depends on lane depths as well as the withdrawal rate of individual products. Therefore, it might be beneficial to store different classes of products in different lane depths. A careful determination of the lane depths is necessary in order to achieve the best storage space utilization. Besides lane depths, the pallet block-stacking problem also determines such decisions as aisle width and orientation, stack height, and storage clearance to achieve a balanced tradeoff between storage space utilization and material handling efficiency. Moder and Thornton (1965) propose ways of stacking pallets in a warehouse and discuss their influence on space utilization and ease of storage and retrieval. They consider such factors as lane depth, pallet placement angle with regards to the aisle, and spacing between storage lanes. Berry (1968) develops analytic models to evaluate the total space requirement and the average travel distance for given block-stacking patterns with different aisle configurations, lane depths, throughput rates, and number of SKUs. Marsh (1979, 1983) uses simulation to evaluate the effect of alternate lane depths and SKU-to-lane assignment rules on space utilization. Goetschalckx and Ratliff (1991) develop an efficient dynamic programming algorithm to maximize space utilization by selecting lane depths out of a limited number of allowable depths and assigning incoming shipments to the different lane depths. Larson et al. (1997) propose a three-step heuristic for the block-stacking problem of class-based pallet storage with the purpose to maximize storage space utilization and minimize material handling cost.

A storage department also can be divided into zones for organizing order picking operations. The fundamental advantages of zone picking are the limited space the picker has to travel to pick an order, the increased familiarity of the picker with a subset of the SKUs, and the reduced order picking time span for an order if zones are picked in parallel. On the other hand, additional costs may be incurred in zone picking, caused by sorting in parallel zone picking and by the queuing in sequential zone picking. Storage needs to be planned for zone picking to determine the specification (the number, size, and shape) of the zones and to assign SKUs to zones in such a way that minimizes the total order picking cost and balance the workloads across zones. The literature on the storage planning for zone picking is very limited. Gray et al. (1992) present a hierarchical framework for designing warehouses with zone picking to determine the number of zones and pickers, zone sizes (storage spaces per zone), storage assignment across and within zones, and order batch size. The effects of zone shape (i.e., the number of aisles per zone and the length of aisles) on operational cost is investigated by Petersen (2002) with simulation. It is shown that zone shape has a substantial impact on the operational cost depending on factors such as the zone size and the batch size.

2.4.3 Storage location assignment

The storage location assignment problem (SLAP) is to assign incoming products to storage locations in storage departments/zones in order to reduce material handling cost and improve space utilization. Different warehouse departments might use different SLAP policies depending on the department-specific SKU profiles and storage technology. The Storage Location Assignment Problem (SLAP) is formally defined as follows:

Given:

1) information on the storage area, including its physical configuration and storage layout
2) information on the storage locations, including their availability, physical dimensions, and location

3) information on the set of items to be stored, including their physical dimensions, demand, quantity, arrival and departure times

Determine:

The physical location where arriving items will be stored

Subject to performance criteria and constraints such as:

1) storage capacity and efficiency
2) picker capacity and efficiency based on the picker cycle time
3) response time
4) the compatibility between products and storage locations and the compatibility between products
4) item retrieval policy such as FIFO (first-in, first-out), LIFO (last-in, first-out), BFIFO (batch first-in, first-out). When using the BFIFO policy, items that arrived in the same replenishment batch are considered to be equivalent.

In typical warehouse operations, the physical storage infrastructure and its characteristics are known when planning the storage location assignment. The availability of storage locations is always known in automated warehouses and often known in mechanized warehouses. The storage assignment problem can be divided into three classes depending on the amount of information known about the arrival and departure of the products stored in the warehouse: 1) item information, 2) product information, or 3) no information. Different operational policies exist for each of these classes, and their implementation and performance have been discussed extensively in the literature. Most of the research has focused on unit-load warehouses. Of course, these SLAP policies can be applied to non unit-load warehouses as well, but it is usually much more difficult to provide analytical results because of the complexity of computing the associated material handling times and cost involved in a non unit-load warehouse (for example when batching and routing are used).

Storage Location Assignment Problem based on Item Information (SLAP/II)

In the SLAP/II problem, it is assumed that complete information is known about the arrival and departure time of the individual items. It is very unlikely that information on individual items will be available in typical warehousing operations, but it may be available in the case of short term planning of container ports or airport gates. The resulting problem is a specially structured Assignment Problem (AP), where items are assigned to storage locations. The special structure derives from the property that two items can occupy the same storage location, provided they do not occupy it at the same time. This problem has been called the Vector Assignment Problem (VAP), since the occupation is no longer expressed as a single binary status variable but as a vector over the different time periods (Goetschalckx (1998)). The optimal solution of this problem for typical warehousing operations is computationally impractical because of the very large problem instances. The problem is of interest to academic research in warehouse operations because it provides a cost lower bound or performance upper bound. An example of a heuristic SLAP/II policy is the Duration-of-Stay (DOS) policy of Goetschalckx and Ratliff (1990). In DOS-based storage policies, the expected DOS of the i^{th} unit of a SKU with replenishment lot size Q is i/λ for $i = 1, 2, \ldots, Q$, where λ is the demand rate of that SKU. Then the items of all the different products having the shortest DOS are assigned to the closest locations. Hence, the items of a single replenishment batch of a single product are not stored together in the warehouse. Under some unrealistic assumptions on the scheduling and size of product replenishments, it can be shown that the DOS storage policy is optimal for both material handling effort and required storage capacity (see also Thonemann and Brandeau (1998)). In practice, DOS-based policies are difficult to implement since it requires the tracking and management of each stored unit in the warehouse. Also the performance of DOS-based policies depends greatly on factors such as the skewness of demands, balance of input and output flows, inventory control policies, and the detailed implementations. Kulturel et al. (1999) compare class-based storage and DOS-based storage using simulation and show that the former consistently outperforms the latter in practical settings.

Storage Location Assignment Problem based on Product Information (SLAP/PI)

Often only product information is known about the items to be stored, and items are instances of products. Products may be classified into product classes. The assignment problem now assigns an individual item to a product class based on its product characteristics, and assigns a product class to storage locations. The location of an item in its class is most often done using some simple rule, such as nearest location, or randomly. If the number of classes is equal to the number of products, then this policy is called Dedicated Storage. If the number of classes is equal to one, this policy is denoted as Random Storage. In real-life warehousing operations, a small number of classes ranging from 3 to 5 are used. This policy is called Class-Based Storage.

Different criteria can be used to assign a product (class) to storage locations. The three most frequently used criteria are: 1) demand rate, 2) maximum inventory, and 3) turnover. For the demand rate policy, products are ranked by decreasing demand rate and the classes with high demand rate are assigned the most desirable locations. For the maximum inventory policy, products are ranked by increasing maximum inventory, which is the sum of their safety inventory and replenishment (cycle) inventory. The storage classes with the lowest maximum inventory are assigned the most desirable locations. For the turnover policy, products are ranked by the ratio of their demand rate divided by their maximum inventory. Products with the highest turnover are stored in the most desirable locations. The turnover policy is the most comprehensively studied one in the literature.

The turnover-based policy for dedicated storage was first described by Heskett (1963, 1964) as the Cube-per-Order index (COI) rule without a proof of its optimality. Kallina and Lynn (1976) discussed the implementation of the COI rule in practice. The COI rule is easy to implement and has the intuitive appeal of locating compact, fast-moving items in readily accessible locations. Furthermore, the COI rule is proved to be optimal for dedicated storage when the following assumptions are satisfied:

(1) The objective is to minimize the long-term average order picking cost.

(2) The travel cost depends only on locations. Examples that do not satisfy this assumption include the case when the travel cost is item dependent or when there are multiple I/O points, and products have different probability of moving from/to the I/O points, i.e., it does not satisfy the factoring assumption as defined in Mallette and Francis (1972).

(3) When dual or multi-command order picking is used, there is no dependence between the picked items in the same picking tour.

(4) Certain routing policies are assumed for multi-command order picking, e.g., Jarvis and McDowell (1991) assume using the traversal routing policy for the conventional multi-aisle order picking system.

(5) There are no compatibility constraints that limit the storage location assignment, e.g., certain items must and/or cannot be put together.

Tables 2.2 and 2.3 give a summary of the literature on dedicated storage assignment policies. Table 2.2 classifies the papers that discuss the COI rule and its variants for different order picking systems and its optimality based on the above assumptions; Table 2.3 gives other heuristic algorithms for dedicated storage when some of the previous assumptions are not satisfied, and therefore the COI rule is not directly applicable.

The turnover-based policy for class-based storage is first discussed by Hausman et al. (1976), Graves et al. (1977), and Schwarz et al. (1978). They compare randomized storage, dedicated storage, and class-based storage in single-command and dual-command AS/RSs using both analytical models and simulations. The results show that the turnover-based policy for class-based storage with relatively few classes could achieve good performance in terms of both material-handling cost and storage capacity.

Table 2.2 COI-based dedicated storage assignment policies

	Single-Command	**Dual-Command**	**Multi-Command**	**Carousel**
COI rules and its variants	Mallette and Francis (1972) Harmatuck (1976)	Malmborg and Krishnakumar (1987) Malmborg and Krishnakumar (1990)	Malmborg and Krishnakumar (1989) Jarvis and McDowell (1991)	Bengu (1995) Vickson (1996)

Table 2.3 Other dedicated storage assignment policies

Citation	Problem Summary	Algorithm
Montulet et al. (1998)	The objective is to minimize the peak operations cost.	Branch and Bound
Lee (1992) Rosenwein (1994) Brynzer and Johansson (1996) van Oudheusden and Zhu (1992)	Items are not independent such that some items are more likely to appear on the same order.	Cluster analysis; Space filling curve based heuristics
Malmborg (1995)	All items of any SKU must be located in the same aisle in a multi-aisle AS/RS system.	Random search plus simulated annealing
Lai et al. (2002) Zhang et al. (2000) Zhang et al. (2002)	Storage location assignment is constrained by product size; all items of the same product must be placed at adjacent locations; and travel costs are item dependent.	Simulated annealing; Genetic algorithms

The implementation of class-based storage (i.e., the number of classes, the assignment of products to classes, and the storage locations for each class) has significant impact on the required storage space and the material handling cost in a warehouse. Research on this problem has been largely focused on AS/RS, especially single-command AS/RS. Hausman et al. (1976) show that for single-command AS/RS with the Chebyshev metric, the ideal shape of storage regions is L-shaped. For such systems, the problem reduces to determining the number and boundaries of the classes. Explicit analytical solutions for the class boundaries can be derived for the case with 2 or 3 classes, as shown by Hausman et al. (1976), Kouvelis and Papanicolaou (1995), and Eynan and Rosenblatt (1994). For the general *n*-class case, Rosenblatt and Eynan (1989) and Eynan and Rosenblatt (1994) suggest a one-dimensional search procedure to find the optimal boundaries. The implementation of class-based storage in multi-command AS/RS is discussed in Guenov and Raeside (1992).

Dedicated storage requires more storage space than class-based storage since sufficient storage locations have to be reserved for the maximum inventory of each product, and therefore increases warehouse space cost and material handling cost. On the other hand, dedicated storage has the advantage that the control of the warehouse is very simple since items of a product will always be stored in the same locations and sufficient space is always available for all the items in replenishment batches. The simplicity advantage is decreasing in importance because the introduction of information technologies such as WMS, bar coding, and radio frequency tags provides a real-time accurate inventory map of the warehouse. The advantages of robustness and simplicity of dedicated storage must be traded off against the increased required storage space and material handling cost.

Storage Location Assignment Problem based on No Information (SAP/NI)

If no information is available on the characteristics of the arriving items, only very simple storage policies can be constructed. In this case the most frequently used policies are 1) Closest-Open-Location (COL), 2) Farthest-Open-Location (FOL), 3) Random (RAN), and 4) Longest-Open-Location (LOL). The first two policies pick an open location based on its distance to the receiving dock; the last policy picks the location that has been vacant for the longest time. It is not known if there is any significant performance difference between them.

SLAP Summary

Most of the SLAP research has focused on the case of Unit Load systems operating under Product Information. The turnover or COI policy has been shown to be optimal for the case with restrictive assumptions such as single command, dedicated storage, and product-independent travel costs. However, simulation typically has been used to show that the turnover policy nearly always performs the best in more general cases.

All of the above research on SLAP assumes that replenishment lot sizes of the SKUs are given. However, Wilson (1977) demonstrates that the lot sizing problem and the SLAP should be considered simultaneously in order to achieve an optimal total cost including both inventory cost and material handling cost. Algorithms for the integrated lot sizing and SLAP problem can be found in Wilson (1977), Hodgson and Lowe (1982), Malmborg et al. (1986), Malmborg and Deutsch (1988), and Malmborg et al. (1988).

The version of the SLAP problem studied in the literature is most often static, i.e., it assumes that the incoming and outgoing material flow patterns are stationary over the planning horizon. In reality, the material flow changes dynamically due to factors such as seasonality and the life cycles of products. Therefore, the storage location assignment should be adjusted to reflect changing material flow requirements. One possibility is to relocate those items whose expected retrieval rate has increased (decreased) closer to (farther from) the I/O point. Such relocations are only beneficial when the expected saving in order picking outweighs the corresponding relocation cost. Therefore, decisions must be made carefully concerning which set of items to be relocated, where to relocate them, and how to schedule the relocations. Another type of relocation might take place as a result of the uncertainty in incoming shipments. For example, Roll and Rosenblatt (1987) describes the situation when the storage area is divided into separate zones and any incoming shipment must be stored within a single zone. It might happen that none of the zones has sufficient space to accommodate an incoming shipment. In such cases, it is advisable to free some space in a certain zone to accommodate the incoming shipment by shifting some stored products in that zone to other zones. Table 2.4 gives a summary of the literature on the various dynamic storage location assignment problems.

Table 2.4 Dynamic storage location assignment problems

Citation	Problem statement	Method
Christofides and Colloff (1972)	The set of items to be relocated and their destinations are given, and the problem is to route the relocation tour to minimize the total relocation cost.	Two-stage heuristics that is optimal in a restricted case.
Muralidharan et al. (1995)	The set of high-demand items to be relocated and their destinations are given, and the problem is to route the relocation tour to minimize the total relocation cost.	A nearest-neighbor heuristic and an insertion heuristic
Jaikumar and Solomon (1990)	Determine the items to be relocated and their destinations with the objective to find the minimum number of relocations that results in a throughput satisfying the throughput requirement in the following busy periods.	Optimal ranking algorithm
Sadiq et al. (1996)	Determine the relocation schedule in face of the dynamically changing order structure, i.e., relocate items that are more likely to appear in the same order in clusters.	Rule of thumb procedure based on cluster techniques
Roll and Rosenblatt (1987)	Using zone storage without splitting, it might happen that none of the zones has sufficient space to accommodate an incoming shipment. The problem is how to shift some stored products in a certain zone to other zones in order to free space for the incoming shipment.	Rule of thumb procedure

2.5 Order picking

Different order picking methods can be employed in a warehouse, for example, single-order picking, batching and sort-while-pick, batching and sort-after-pick, single-order picking with zoning, and batching with zoning (Yoon and Sharp (1996)). Each order picking method consists of some or all of the following basic steps: batching, routing and sequencing, and sortation.

2.5.1 Batching

The batching problem is part of planning for order picking. Orders are received and subsequently released for fulfillment. Given a set of released orders, the problem is to partition the set into batches, where each batch will be picked and accumulated for packing and shipping during a specific time window, or “pick wave.” The time required to pick the items in any batch should not exceed the time window or pick wave duration. If zone picking is employed, the batch should balance pick effort across the zones to achieve high picker utilization, while minimizing pick time so that the number of pickers required is minimized.

The batching problem can be stated as:

Given: warehouse configuration, pick wave schedule, and a set of orders to pick during a shift

Find: a partition of orders for assignment to waves and pickers

Criteria: picker effort, imbalance among pickers

Constraints: time slots, picker capacity, order due dates

In creating an abstract statement of the problem, there are potentially two levels of partitioning: (1) partitioning in time (into pick waves); and (2) partitioning among pickers in a wave or zone. Constraints include the picker capacity during the time interval associated with a pick wave, and time constraints on when an order should be completed.

Partitioning into time slots is essentially a “bin packing” type problem, where the goal is to balance the pick time among the time slots or pick waves. The difficulty, of course, is that the time required to pick a batch is not known until the batch has been determined, partitioned among individual picker, and the pickers have been routed through the warehouse.

Partitioning of the orders among the pickers is a variation of the classical vehicle routing problem (VRP), in which “stops” are assigned to routes and the objective is to minimize the total route distance or time. However, in the order-batching problem, assigning an order to a picker's route implies that all the picking locations for the SKUs in this order are assigned to this route. This is similar to the pick-up and delivery vehicle routing problem, or the dial-a-ride problem, where a service request consists of a pick-up location and a drop-off location with time precedence. In the order partitioning problem, there may be many stops (SKUs) associated with a single service request (order) but there are no precedence constraints.

The published research has focused primarily on the problem of partitioning among pickers. There are two major types of batching heuristics that attempt to minimize total picking effort and are based on VRP heuristics. A seed algorithm selects initially a single seed order in the batch. More orders are then added according to a route closeness criterion until no more orders can be added due to a capacity constraint. The capacity constraint can be based on total pick time, number of orders in the batch, or weight. A savings heuristic starts by assigning each order to a separate batch. The algorithm then iteratively selects a pair of batches to be combined based on the savings of combining them until no more batches can be combined due to the capacity constraint.

Central to both types of algorithms is an order-to-route closeness metric, which defines the order addition rule in the seed algorithms and the combination rule in the saving algorithms. Table 2.5 summarizes closeness metrics proposed in the literature. The seed and savings algorithms proposed in the literature are similar in terms of their general procedure, but differ in the closeness metric used. Table 2.6 shows the different algorithms and the closeness metrics they used as shown by the bold number after each citation.

Table 2.5 Order closeness metrics for batching

Index	Closeness Metric	Example
1	Number of common locations between two orders	Elsayed (1981)
2	Combined number of locations of two orders	Elsayed and Stern (1983)
3	Sum of the distance between each location of one order and the closest location on the other order	Elsayed and Stern (1983)
4	Difference of the order-theta values of two orders defined based on space-filling curves	Gibson and Sharp (1992)
5	The number of additional aisles to travel when two orders are combined	Rosenwein (1996)
6	Savings in travel when two orders are combined	Elsayed and Unal (1989)
7	Center of gravity metric	Rosenwein (1996)
8	Economic convex hull based metric	Hwang and Lee (1988)
9	Common covered regions or areas	Hwang et al. (1988)

Table 2.6 Order batching heuristics by type

Seed Algorithm		**Saving Algorithm**	
Elsayed (1981)	**(1)**	Rosenwein (1996)	**(5, 7)**
Elsayed and Stern (1983)	**(1, 2, 3)**	Hwang and Lee (1988)	**(8)**
Elsayed and Unal (1989)	**(6)**	Elsayed and Unal (1989)	**(6)**
Gibson and Sharp (1992)	**(3, 4)**	de Koster et al. (1999)	**(6)**
Hwang and Lee (1988)	**(8)**		
Hwang et al. (1988)	**(9)**		
Pan and Liu (1995)	**(1, 3, 4, 6, 8)**		
de Koster et al. (1999)	**(3, 5, 6, 7)**		

Many of the papers listed in Table 2.6 also provide performance evaluation of the different batching algorithms using simulation. It is however difficult to draw general conclusions since the performance depends heavily on factors such as storage location assignment policies, routing policies, the structure of orders, storage systems, and the maximum batch size. A comprehensive study that considers all the above factors and the various batch construction heuristics has not been published at this time. A few results have been published where two policy classes are studied jointly, for example, de Koster et al. (1999) evaluate batching and routing algorithms together, and Ruben and Jacobs (1999) evaluate batching algorithms with different SLAP policies.

Elsayed et al. (1993) present a heuristic for batching orders that have due dates with the objective to minimize earliness and tardiness penalties. Elsayed and Lee (1996) consider batching and sequencing of both storage and retrieval orders such that the total tardiness of the retrieval orders is minimized. Cormier (1987) propose a heuristic for batching and sequencing orders to minimize the weighted sum of order picking time and tardiness in an AS/RS.

Very few papers have developed optimal order batching algorithms. Armstrong et al. (1979) present a mixed-integer formulation for order batching problem in a semi-automated order-picking system with the objective to minimize the total order picking time. The model was solved using Bender's decomposition. Gademann et al. (2001) consider the order batching problem with the objective to minimize the maximum lead time of any of the batches and solve the formulation optimally using a branch-and-bound algorithm.

2.5.2 Sequencing and routing

The sequencing and routing decision in order picking operations determines the best sequence and route of locations for picking and/or storing a given set of items. The objective is typically to minimize the total material handling cost. This problem is a warehouse-specific Traveling Salesman Problem (TSP), where the picking/storing location of an item is given. The problem where there are several candidate locations for the retrieval or storage of an item is much more complex and few research results are available, although it is often found in practice. The TSP problem in the warehouse is special because of the aisle structure of the possible travel paths. The published research focuses on four classes of warehouse systems, i.e., conventional multi-parallel-aisle systems, man-on-board AS/RS systems, unit-load AS/RS systems, and carousel systems.

Sequencing and routing for conventional multi-parallel-aisle systems

In a conventional multi-parallel-aisle system, the aisle structure limits the TSP state space, which greatly simplifies its solution. Ratliff and Rosenthal (1983) propose a polynomial-time dynamic programming algorithm to optimally solve this problem. The algorithm depends on the following assumptions: parallel, narrow and equal aisles, a single I/O point for the picker in the warehouse, the aisles connected by a cross aisle at each end, and the SKU locations given. Other authors have relaxed some of these assumptions and their results are summarized in Table 2.7. Hall (1993) compares the performance of several simple heuristics for the multi-parallel-aisle systems, such as the traversal and return policies, through analytical models. Petersen (1997, 1999) provide a similar study through simulation.

Table 2.7 Routing algorithms for conventional multi-parallel-aisle warehouses

Citation	Problem Setting	Algorithm	Optimal or not
Ratliff and Rosenthal (1983)	1. Narrow aisles; 2. A tour starts and ends at the central depot 3. Only two cross aisles located at the ends of picking aisles; 4. Picking locations are given	A dynamic programming based algorithm	Optimal with computational time linear in the number of aisles
Goetschalckx and Ratliff (1988b) Goetschalckx and Ratliff (1988a)	Routing in wide aisles	A shortest path algorithm and a set-covering based algorithm with the consecutive ones property	Optimal for the routing within a single aisle
de Koster and van der Poort (1998)	A tour can start and end at the head of any picking aisle	An extension of Ratliff and Rosenthal (1983)	Optimal
Roodbergen and de Koster (2001b)	There are three cross aisles	An extension of Ratliff and Rosenthal (1983)	Optimal
Vaughan and Petersen (1999) Roodbergen and de Koster (2001a)	There are arbitrary number of cross aisles	Dynamic programming based heuristics	Heuristics
Daniels et al. (1998)	Picking locations need to be selected before routing	TSP based heuristics with local search methods	Heuristics

Sequencing and routing for man-on-board AS/RS

The routing problem for man-on-board AS/RS is a TSP problem with a Chebyshev distance metric. The literature on this problem has been primarily focused on efficient heuristics. Gudehus (1973) describes the band heuristic, which divides the rack into two equal height horizontal bands; the points in the lower band are visited in the increasing x-coordinate direction, while the points in the upper band are visited in the opposite direction. If the tour must visit many points, the rack may be divided into several pairs of horizontal bands. Goetschalckx and Ratliff (1988c) propose a convex hull algorithm based on the property of Chebyshev metric that some points not on the convex hull can be inserted into it without incurring additional travel distance. The algorithm constructs the convex hull of all the picking locations, then those free insertion locations for each segment of the convex hull are identified and inserted into the convex hull, and then the remaining points are sequentially inserted into the tour in a way that minimizes the increase in tour length for each insertion. The band algorithm is easy to implement and computationally efficient, but might give inferior solutions in some cases. On the other hand, the convex hull algorithm is effective in finding short tours, but is difficult to implement (to find the convex hull and free insertion points) and less computationally efficient.

Bozer et al. (1990) propose the ½ band insertion heuristic, which is a combination of the band and convex hull heuristics. The heuristic first divides the rack into three equal width horizontal bands, all the points in the first and third band are routed in the same way as in the band heuristic to obtain a partial tour, and the points in the middle band are then inserted as in the final stage of the convex hull algorithm. Other heuristics in the literature include the center sweep heuristic (Bozer et al. (1990)), the space-filling curve based heuristic (Bartholdi and Platzman (1988)), and the combined convex hull heuristic for a variation of the man-on-board systems (Hwang and Song (1993)). Local improvement procedures (Bozer et al. (1990), Makris and Giakoumakis (2003)) can be used together with all the above heuristics to further reduce the tour length.

Bozer et al. (1990) give a comprehensive comparison of these heuristics, and conclude that the convex hull and ½ band insertion heuristics consistently outperform the others, and suggest the use of the ½ band heuristic because it achieves performance close to that of the convex hull algorithm, but is very simple to implement and runs very efficiently. Bachers et al. (1988) provide a comparison of several traditional TSP heuristics, such as the nearest-neighbor method, the successive insertion method, and the local search method, through simulation.

Sequencing and routing for unit-load AS/RS

The routing problem for unit-load AS/RS (also called the interleaving problem) pairs a storage operation with a retrieval operation for a dual command cycle. Graves et al. (1977) demonstrate that careful interleaving can effectively reduce the total travel distance by reducing the unproductive travel between storage and retrieval locations. The algorithms reported in the literature are either static or dynamic. Static algorithms fix a block of storage and retrieval requests, sequence the requests in the block, and execute the resulting schedule ignoring new storage and retrieval requests. Dynamic algorithms re-sequence the storages and retrievals whenever new requests arrive. The static sequencing problem for randomized and class-based storage is believed to be NP-hard, and most algorithms for this problem use a nearest-neighbor heuristic or one of its variations. Han et al. (1987) proposed a match of a storage location with a retrieval location that has the minimum travel distance between them. Lee and Schaefer (1996) developed an assignment formulation and can find an optimum or near-optimum solution for problems of moderate size. The static case for dedicated storage policies can be solved in polynomial time by formulating it as a transportation or assignment problem (van den Berg and Gademann (1999), Lee and Schaefer (1997)). Table 2.8 summarizes the static algorithms for different systems and storage policies. Dynamic algorithms in the literature are mainly direct extensions of the static algorithms that re-sequence the requests whenever a new request arrives in the system as reported by Lee and Schaefer (1997), Eben-Chaime (1992), and Ascheuer et al. (1999). Seidmann (1988) proposes a different dynamic control approach based on artificial intelligence techniques.

Table 2.8 Static sequencing algorithms for dual-command AS/RS

	Citation	Problem Setting	Algorithm	Optimal or not
Randomized Storage	Han et al. (1987)	Unit-load AS/RS	Nearest-neighbor heuristic	Heuristic
	Lee and Schaefer (1996)	Unit-load AS/RS	Assignment-based algorithm	ε-optimal
	Mahajan et al. (1998)	Miniload end-of-aisle AS/RS	Nearest-neighbor heuristic	Heuristic
	Keserla and Peters (1994)	Unit-load dual shuttle AS/RS	Minimum-perimeter heuristic	Heuristic
	Sarker et al. (1991)	Unit-load dual shuttle AS/RS	Nearest-neighbor heuristic	Heuristic
Dedicated Storage	van den Berg and Gademann (1999)	Unit-load AS/RS	Transportation problem	Optimal
	Lee and Schaefer (1997)	Unit-load AS/RS	Assignment problem	Optimal
Class-based Storage	Eynan and Rosenblatt (1993)	Unit-load AS/RS	Nearest-neighbor heuristic	Heuristic
	Sarker et al. (1994)	Unit-load dual shuttle AS/RS	Nearest-neighbor heuristic	Heuristic

In some cases, Just-In-Time performance of the AS/RS is more important than minimizing the total operational cost. For example, if the AS/RS is used to feed a production line, it is important that the requested materials are retrieved at the time determined by the production schedule. Lee and Kim (1995) and Linn and Xie (1993) develop heuristics to sequence the storage and retrieval requests in order to improve the due date related performance.

Several authors have studied the dwell point selection problem in a unit-load AS/RS. The dwell point is the position where the S/R shuttle stops when the system is idle. The dwell point can be selected to minimize the expected travel time to the position of the first transaction after an idle period, and thus improve system response. Bozer and White (1984) describe some rules-of-thumb to determine the dwell point. Egbelu (1991) and Chang and Egbelu (1997) present LP models to find the optimal dwell point that minimizes the expected response time and the maximum response time respectively. Hwang and Lim (1993) presents more efficient algorithms to solve the models proposed by Egbelu (1991) based on the facility location formulation. Peters et al. (1996) and van den Berg (2002) provide closed form solutions for the optimal dwell point to minimize the expected response time using analytic models based on continuous approximation of the storage rack. Egbelu and Wu (1993) use simulations to evaluate the performance of the LP-based rules in Egbelu (1991) and the rules-of-thumb in Bozer and White (1984) in practical environments, and find that the former outperforms the latter in most cases, especially when the system uses dedicated storage and is not very busy.

Simulation studies of the operational policies for an unit-load AS/RS can be found in Linn and Wysk (1987) and van den Berg and Gademann (2000), which compare different sequencing rules, dwell point selection rules, and storage location assignment rules under various conditions of the product mix and the traffic intensity.

Sequencing and routing for carousel systems

The sequencing problem in carousel systems was first considered by Bartholdi and Platzman (1986). They assume that the orders are picked one at a time, which leads to two sequencing problems, i.e., the pick sequencing within an order and the sequencing of orders. The effect of the latter is not significant when the order arrival rate is small compared with the order retrieval rate, so the problem simplifies to the pick sequencing within the orders. They present a polynomial algorithm to optimally solve this problem, as well as some simple heuristics that are easier to compute and perform well when the number of picks is large relative to the total storage space. When the order arrival rate is large, the sequencing of orders must be considered in order to minimize the unproductive time of traveling from the end position of one order to the start position of the next. In this case, an efficient heuristic is proposed based on the additional assumption that each order is picked along its shortest spanning interval, which is the shortest interval that covers all the picking locations of the order. It is shown that the proposed heuristic will produce a solution that is never more than 1 revolution longer than the optimal, i.e., the more orders to be picked, the better the solution.

Ghosh and Wells (1992) and van den Berg (1996) consider the problem when the sequence of orders is fixed (but the pick sequence within the orders are free), and propose efficient dynamic programming approaches to optimally solve it. van den Berg (1996) also considers the case when both in-order and between-order picking sequences are free by assuming that each order is picked along its shortest spanning interval. They formulate this problem as a Rural Postman Problem on a circle and solve it to optimality. Furthermore, they show that the solution obtained with the extra constraint is at most 1.5 revolutions more than the optimal without the extra constraint. The above research treats the carousel as a one-dimensional system, i.e., the travel perpendicular to the rotation of the carousel was not considered. Wen and Chang (1988) consider a two-dimensional carousel system and propose three heuristics that are extensions of Bartholdi and Platzman's optimal algorithm. Han and McGinnis (1986) and Han et al. (1988) extend the nearest-neighbor heuristics discussed earlier for the dual-command AS/RS to carousels and rotary racks (A rotary rack is similar to a carousel except that it has several layers, and each layer can be operated independently).

In summary, the sequencing and routing problem is the most studied problem in warehouse operation. Most of the research assumes that the locations to be visited are given. The problem when multiple candidate locations are available for the retrieval or storage of an SKU remains an interesting and challenging research problem. Also, in a warehouse setting, batching is closely related to sequencing, and therefore those problems require a joint solution method. Finally, because of the confined and narrow travel paths in a warehouse, another relevant variant of the sequencing and routing problem would consider congestion when there are multiple order picking tours executed at the same time period in the same area.

2.5.3 Sortation

Sorting is required when multiple orders are picked together. It can be performed either during the picking process (sort-while-pick) or after the picking process (sort-after-pick). Sort-while-pick is quite straightforward and is typically modeled by inflating the item extraction time. For sort-after-pick, a separate downstream sortation system is used to perform the sorting function. A number of questions are related to the operation of the sortation system.

Sortation systems used in warehouses usually include an accumulation conveyor, a recirculation conveyor, and sortation exit lanes, and they operate simultaneously on all the orders in a single pick-wave. Items for a pick wave arrive at the accumulation conveyor where they wait to be released into the sortation process. They are put onto the recirculation conveyor through an induction point after the items in the previous pick-wave finish their sorting process (in some cases, the items are allowed to enter the recirculation conveyor before the previous wave has totally finished its sorting). The orders are assigned to sortation lanes according to order-to-lane assignment rules. Items circulate in the recirculation conveyor and enter the assigned sortation lane if all items of the preceding order assigned to that lane have been sorted. If not, the items bypass the sortation lane and re-circulate. Eventually, sorted orders are removed from sortation lanes, checked, packed, and shipped. Therefore, the operation problem for sortation involves decisions such as wave-releasing and order-to-lane assignment so that the orders can be efficiently sorted in a given wave.

There are relatively few research results in this area. Bozer and Sharp (1985) consider a system that processes a relatively small number of large orders. In this case, each sortation lane is typically dedicated to one order. The authors use simulation to analyze the dependence of the system throughput on factors such as the induction capacity, the number of lanes, and the length of lanes. Bozer et al. (1988) consider a similar problem but with a large number of small orders. In this case, each lane is assigned several orders and an order-to-lane assignment policy determines how and when the orders enter the sortation lanes. Orders that are not yet assigned a lane are forced to recirculate. Using simulation, they compare different order-to-lane assignment rules, which include the simplest FCFS rule and priority rules based on the sizes of orders or the time that an order has been in the system. They find that the FCFS rule consistently outperforms more elaborated rules. Johnson (1998) verifies this result with analytical models for the sortation system operated under different order-to-lane assignment rules. Meller (1997) propose an optimal order-to-lane assignment method to minimize the sortation time for a pick-wave based on a set-partitioning model.

In practice, the sortation time in an automatic sortation system might not be a critical factor as long as all orders can be sorted within a given wave. Therefore, simple heuristics would suffice in most practical cases if orders were partitioned into pick waves in a balanced way.

2.6 Conclusions and discussions

The distribution of the research results among the various warehouse operational problems is shown in Figure 2.1, where the number in parentheses represents the number of papers addressing the corresponding problem. It is clear that the past research has focused strongly on storage and order picking. This is not surprising since these are the two warehouse functions that have the largest impact on the overall warehouse operational performance including storage capacity, space utilization, and order picking efficiency.

On the other hand, the development of research is not well balanced. Some problems received far more attention from the research community than others. For example, the SLAP and routing problems account respectively for 32% and 38% of the total surveyed literature, while zoning accounts for less than 6%. Furthermore, there is little evidence of collaboration between the academic research community and industry. Many of the research results are not sufficiently communicated to industry to make a significant impact on the practice of warehouse operations. More communication from both sides might help to better identify the real challenges faced in warehouse operations, to appreciate the opportunities for better operation, and to realize these opportunities by close cooperation between researchers and practitioners.

The problems discussed in this chapter are at the operational level, which means that decisions need to be made quite frequently and the influence of these decisions is typically of a short duration and localized. Such decisions typically need to be made quickly without extensive computational resources. This tends to encourage the use of heuristic procedures that can reliably find a good solution in a reasonable amount of time. In addition, from the management point of view, an ideal solution method should be simple, intuitive, and reliable so the training costs in the warehouse are minimized as much as possible.

Another consequence of the operational nature of the problems discussed in this paper is that the problems should be considered dynamically by constantly incorporating new information about the operating environments. Some research on the dynamic planning of warehouse operations exists, but the dynamic problems are much less studied than the static variants. Furthermore, research in the literature usually concentrates on certain standard performance measures, such as the total order picking cost. In many practical situations, different objectives such as the tardiness, or the order cycle time, are as important as the traditional aggregate performance measure.

In summary, there continues to be a need for research focusing on the operational management of warehousing systems, where the different processes in the warehouse are considered jointly, the problems are placed in their dynamic nature, and multiple objectives are considered simultaneously. Clearly, the research domain of warehouse operations is very rich and challenging. Given the prevalence of warehouses in the supply chains, such research results can have a significant economic impact.

CHAPTER 3

RESEARCH ON WAREHOUSE DESIGN

3.1 Introduction

This chapter reviews the research on warehouse design, performance evaluation, case studies, and computational support tools. Warehouse design consists of five major activities (Figure 2.1): determining the overall structure; sizing and dimensioning of the warehouse and its departments; determining the detailed layout within each department; selecting warehouse equipment; and selecting operational strategies. The overall structure (or conceptual design) determines the material flow pattern within the warehouse, the specification of functional departments, and the spatial relationship of departments. The sizing and dimensioning problem determines the size and dimension of the warehouse as well as the space allocation among warehouse departments. Department layout is the detailed configuration for warehouse departments, for example, aisle configuration in the retrieval area, pallet block-stacking pattern in the reserve storage area, and configuration of the AS/RS. The equipment selection problem is to determine an appropriate automation level for the warehouse, and specify specific equipment types for storage, transportation, order picking, sortation, etc. The operation strategy selection problem is to determine how the warehouse is going to be operated, for example, with regards to storage and order picking. Here operation strategies refer to those decisions that have global effects on other design decisions, and therefore need to be considered in the design phase. Examples of such operation strategies include using randomized storage or dedicated storage, using zone picking or not, and using sort-while-pick or sort-after-pick.

Detailed operational policies, such as how to batch and route the order picking tour, are not considered design problems here, and therefore are discussed in chapter 2. It should be emphasized that the above design problems are strongly coupled, and it is difficult to define a sharp boundary between them. Therefore, the above classification should not be regarded as unique, nor does it imply any of the problems should be solved independently without regarding the other problems. Furthermore, one should not ignore operational problems in the design phase since operational efficiency is strongly affected by the design decisions and it can be very expensive or impossible to change the design decisions once the warehouse has been constructed.

Performance evaluation is important for both warehouse design and operation in the sense that it assesses the performance of a warehouse in terms of cost, throughput, space utilization, and service to provide feedback about how a specific design or operational policy performs compared with the requirements, and how it can be improved. Furthermore, a good performance evaluation model can help the designer to quickly evaluate many design alternatives and narrow down the design space during the early design stage. Performance evaluation methods include benchmarking, analytical models, and simulation models. This review will mainly focus on the former two. However, this should not obscure the fact that simulation is still the most widely used technique in the academic literature as well as in practice.

Some case studies and computational systems are also discussed in this chapter. Research in these two directions is very limited. However, it is our belief that more case studies and computational tools for warehouse design and operation will help us to bridge the gap between academic research and practical application, and therefore, need to be further developed in the future.

The next four sections will discuss the literature on warehouse design, performance evaluation, case studies, and computational systems respectively. The final section gives some conclusions and future research directions.

3.2 Warehouse design

3.2.1 Overall Structure

The overall structure (or conceptual design) of a warehouse involves tasks such as material flow modeling, functional department specification, and spatial relationship specification of departments within the warehouse. At this stage, the designer develops a preliminary design plan considering requirements for capacity, throughput, budget, and space. In order to evaluate different preliminary design alternatives, the designer needs specific (although not exact) knowledge about the size of the warehouse, possible material handling equipment, and possible operational policies. The methods to perform such evaluations are mainly based on rough rule-of-thumb calculations.

Park and Webster (1989) present a procedure to select a preliminary design for a unit-load warehouse among different alternatives that are combinations of alternative equipment types, storage rules, and order picking policies. The initial investment cost and annual operational cost for each alternative is estimated using simple analytic equations, and the best alternative that satisfies all design requirements is chosen. Gray et al. (1992) propose a multi-stage hierarchical approach that determines system configuration and equipment selection, storage allocation and location assignment, and operation policies in a sequential and iterative way. Simple calculations are employed to evaluate the tradeoffs and prune the design space to a few superior alternatives. Simulation is then used to provide detailed performance evaluation of the resulting alternatives. Yoon and Sharp (1996) propose a structured procedure for the design of order picking systems, which includes stages such as design information collection, design alternative development, and performance evaluation. Each stage consists of a set of sub-problems, for example, the design alternative development stage includes specification of equipments, specification of operating strategies, physical transformation of items, and information transformation.

In summary, research in the overall warehouse structure design is very limited. The methodologies discussed above are all similar in the sense that they divide the complex design problem into a set of simpler sub-problems, which are then solved in a sequential and iterative way to develop the design alternatives. The resulting design is not detailed enough, and needs to be further refined to determine the detailed design.

3.2.2 Sizing and dimensioning

Warehouse sizing and dimensioning determines the size and dimension of the warehouse and its departments, which has important implications on costs such as construction cost, inventory holding and replenishment cost, and material handling cost.

Assuming that the warehouse has no control over inventory, warehouse sizing determines the appropriate storage capacity in order to satisfy the stochastic demand for storage space. White and Francis (1971) studied this problem for a single product over a finite planning horizon. Costs considered include those due to warehouse construction, storage of products within the warehouse, and storage demand not satisfied by storage in the warehouse. Problems with either fixed or changeable storage size are modeled. In the first model, the problem is to determine the optimal fixed storage size and a simple procedure is proposed to find the optimum. The second model allows changes in the storage size over the planning horizon (e.g. by leasing additional storage space), so the decision variables are the storage sizes for each time period. A linear programming formulation was presented for the second model, and the optimal solution is found by solving a network flow problem. Lowe et al. (1979) give an efficient greedy network flow algorithm for the second problem in White and Francis (1971). Similar problems of determining fixed and changeable warehouse size are also discussed by Hung and Fisk (1984) and Rao and Rao (1998) with different cost formulations.

Levy (1974), Goh et al. (2001), and Cormier and Gunn (1996) consider warehouse sizing problems in the case where the warehouse is responsible for controlling the inventory. Therefore, the cost in their models includes not only warehouse construction cost, but also inventory holding and replenishment cost. Levy (1974) presents analytic models to determine the optimal storage size for a single product with either deterministic or stochastic demand. Goh et al. (2001) find the optimal storage size for both single-product and multi-product cases with deterministic demand. They consider a more realistic piecewise linear model for the warehouse construction cost instead of the traditional linear cost model. Furthermore, they consider the possibility of joint inventory replenishment for the multi-product case, and propose a heuristic to find the warehouse size with a performance bound of 94%. Assuming additional space can be leased to supplement the warehouse, Cormier and Gunn (1996) propose closed-form formulae to determine the optimal warehouse size, the optimal amount of space to lease in each period, and the optimal replenishment quantity for a single product case with deterministic demand. The multi-product case is modeled as a nonlinear optimization problem assuming that there is no staggering of the replenishments. Finally, Rosenblatt and Roll (1988) conduct a simulation study to investigate the dependency of the total required storage capacity on elements such as the reorder point, ordering quantity and demand rate in a stochastic environment.

The warehouse dimensioning problem is first modeled by Francis (1967), who, given a fixed storage size, determines the dimension of the storage department in order to minimize construction and material handling cost. The proposed model is based on a continuous approximation of the storage area without considering aisle structure. Bassan et al. (1980) extends the above work to consider different aisle configurations. Similar to Francis, they also minimize the warehouse construction and operational cost. Rosenblatt and Roll (1984) integrate the optimization model in Bassan et al. (1980) with a simulation model to find the optimal size and dimension of a warehouse that minimizes the total cost (warehouse construction and materials handling cost, storage space shortage cost, management cost due to the use of grouped storage policy). The optimization model is used to determine the optimal dimension for a fixed capacity to minimize the construction cost and materials handling cost, while the simulation model proposed in Rosenblatt and Roll (1988) is used to evaluate the storage shortage cost, which depends on the capacity and number of zones. The linking variables between these two models are the total capacity and number of zones (randomness of storage). Fixing the linking variables, the total cost is obtained by running the sub models. The global optimal solution is achieved by enumerating all the possible combinations of the discretized linking variables. The above work assumes single-command tours in order to evaluate the effect of warehouse dimension on the operational cost, and therefore is not applicable to warehouses that perform multi-command operations.

The above work has been concentrated on the sizing and dimensioning problem assuming the warehouse has a single storage department. In reality, a warehouse might have multiple departments, e.g., the forward picking department, or different storage departments for different classes of SKU. These different departments must be arranged in a single warehouse and compete against each other for space. Therefore, tradeoffs exist in determining the total warehouse size, allocating the warehouse space among departments, and determining the dimension of the warehouse and its departments. Research studying these tradeoffs in the warehouse area is scarce. Pliskin and Dori (1982) propose a method to compare alternative space allocations among different warehouse departments based on multi-attribute value functions, which explicitly capture the tradeoffs among different criteria. Another example is the work by Azadivar (1989), who proposes an approach to optimally allocate space between two departments: one is efficient in terms of storage but inefficient in terms of operation, while the other is the opposite. The objective is to achieve the best system performance by appropriately allocating space between these two departments to balance the storage capacity and operational efficiency tradeoffs.

Another limitation of previous research on warehouse sizing and dimensioning is that they usually assume some basic operational policies, e.g., single-command operations, in order to evaluate the operational cost. However, in reality, which operational policy to be employed in the warehouse is usually not clear at the design phase. Therefore, the designer faces the dilemma of evaluating the operational implications of design decisions without knowing exactly how the warehouse is going to be operated. How to deal with this uncertainty in the design phase remains a difficult and unexplored problem in the warehouse literature.

3.2.3 Department layout

This section discusses layout problems within a warehouse department (mainly the storage department), which are classified as: (P1) pallet block-stacking pattern, i.e., storage lane depth, number of lanes for each depth, stack height, pallet placement angle with regards to the aisle, storage clearance between pallets, and length and width of aisles; (P2) storage department layout, i.e., door location, aisle orientation, length and width of aisles, and number of aisles; and (P3) AS/RS configuration, i.e., dimension of storage racks, number of cranes. These layout problems affect warehouse performances on: (O1) construction and maintenance cost; (O2) material handling cost; (O3) storage capacity, e.g., the ability to accommodate incoming shipments; (O4) space utilization; and (O5) equipment utilization. Each problem is discussed in the literature by different authors considering a subset of the performance measures, which are summarized in Table 3.1.

Table 3.1 A summary of the literature on warehouse layout design

Problem	Citation	Objective	Method	Notes
P1	Moder and Thornton (1965)	O4	Analytical formulae	
	Berry (1968)	O2, O4	Analytical formulae	
	Marsh (1979) Marsh (1983)	O3, O4	Simulation models	
	Goetschalckx and Ratliff (1991)	O4	Heuristic procedure	Mainly on lane depth determination
	Larson et al. (1997)	O2, O4	Heuristic procedure	For class-based storage
P2	Roberts and Reed (1972)	O1, O2	Dynamic Programming	Consider the configuration of storage bays (unit storage blocks)
	Bassan et al. (1980)	O1, O2	Optimal design using analytical formulation	Consider horizontal and vertical aisle orientations, locations of doors, and zoning of the storage area.
	Rosenblatt and Roll (1984)	O1, O2, O3	Optimal two-dimensional search method	Based on Bassan et al's work with additional costs due to the use of grouped storage.
	Pandit and Palekar (1993)	O2	Queuing model	Include not only the ordinary travel time, but also waiting time when all vehicles are busy
P3	Karasawa et al. (1980)	O1, O2, O3	Nonlinear mixed integer problem	The model is solved by generalized Lagrange multiplier method
	Ashayeri et al. (1985)	O1, O2	Nonlinear mixed integer problem	Given rack height, the model can be simplified to a convex problem
	Rosenblatt et al. (1993)	O1, O2, O3	Nonlinear mixed integer problem	System service is evaluated using simulations, if not satisfactory, new constraints are added and the optimization model is solved again to get a new solution
	Zollinger (1996)	O1, O5	Rule of thumb heuristic	
	Malmborg (2001)	O1, O5	Rule of thumb heuristic	A more elaborated variation of Zollinger's rules that consider explicitly operational policies

In the pallet block-stacking problem, a fundamental decision is to determine lane depths to balance the tradeoffs between space utilization and ease of storage/retrieval operations, considering the SKUs' stackability limits, arriving lot sizes, and retrieval patterns. Using deep lane storage could increase space utilization because fewer aisles are needed, but on the other hand could also cause decreased space utilization due to the "honeycombing" effect that results in wasted space unusable for storage of other items before the whole lot is totally depleted from a lane. Honeycombing effect depends on lane depths as well as the withdrawal rate of individual products. Therefore, it might be beneficial to store different classes of products in different lane depths. A careful determination and coordination of the lane depth for different products is necessary in order to achieve the best storage space utilization. Besides lane configuration, the pallet block-stacking problem also determines such decisions as aisle width and orientation, stack height, and storage clearance, which all affect storage space utilization, material handling efficiency, and storage capacity. A number of papers discuss the pallet block-stacking problem. Moder and Thornton (1965) consider ways of stacking pallets in a warehouse and the influence on space utilization and ease of storage and retrieval. They consider such design factors as lane depth, pallet placement angle with regards to the aisle, and spacing between storage lanes. Berry (1968) discusses the tradeoffs between storage efficiency and material handling costs by developing analytic models to evaluate the total warehouse volume (given the storage space requirement) and the average travel distance. The factors considered include warehouse shape, number, length and orientation of aisles, lane depth, throughput rate, and number of SKUs contained in the warehouse. It should be noted that the models for total warehouse volume and models for average travel distance are not integrated, and the warehouse layout that maximize storage efficiency is different from the one that minimizes travel distance. Marsh (1979) uses simulation to evaluate the effect of alternate lane depths and the rules of assigning incoming shipments to lanes on space utilization. Marsh (1983) compares the layout design developed by using the simulation models of Marsh (1979) and the analytic models proposed by Berry (1968). Goetschalckx and Ratliff (1991) develop an efficient dynamic programming algorithm to maximize space utilization by selecting lane depths out of a limited number of allowable depths and assigning incoming shipments to the different lane depths. Larson et al. (1997) propose a three-step heuristic for the layout problem of class-based pallet storage with the purpose to maximize storage space utilization and minimize material handling cost. To summarize, research for the pallet block-stacking problem suggests different rules or algorithms. Some methods give "optimal" results when their assumptions are satisfied. However, the real problem is really complex considering all the different SKUs with different and ever-changing flow activities. It is not clear what method works best in practice, or what is the appropriate method to use in a specific environment.

The storage department layout problem determines the internal layout of a storage department in order to minimize the construction cost and material handling cost. The decisions considered usually include aisle orientations, number of aisles, length and width of aisles, and door locations. In order to evaluate operational costs, some assumptions are usually made about the storage and order picking policies, for example, random storage and single-command order picking are the most common assumptions. Roberts and Reed (1972) assume storage space is available in units of identical bays, and determine the optimal bay configuration to minimize the construction and material handling cost. Bassan et al. (1980) present optimal layout with two different aisle structures in a rectangular warehouse, i.e., the aisles are either parallel or perpendicular to the longitudinal walls. In addition, they also discuss the optimal door locations in the storage department, and the optimal layout when the storage area is divided into different zones. The cost to be minimized is the total construction and material handling cost. Rosenblatt and Roll (1984) extend Bassan et al. (1980) to also include the additional cost due to the use of grouped storage policy. Pandit and Palekar (1993) solve the storage layout problem in order to minimize the expected response time of storage and/or retrieval requests. They propose a queuing model to calculate the total response time including waiting and processing time for different types of layouts. Based on this, an optimization model is solved to find the optimal storage space configurations.

Finally, the AS/RS configuration problem is mainly about determining the number of cranes and aisles, and storage rack dimension in order to minimize construction, maintenance, and operational cost, and/or maximize equipment utilization. The optimal design models or rule-of-thumb procedures summarized in Table 3.1 typically utilize some empirical expressions of the costs based on simple assumptions of operational policies. The AS/RS design problem is discussed by Karasawa et al. (1980), Ashayeri et al. (1985), Rosenblatt et al. (1993), and Malmborg (2001). Karasawa et al. (1980) presents a nonlinear mixed integer model to design automated warehouses. The decision variables are number of cranes, and height and length of storage racks. The model minimizes the total costs including construction and equipment costs while satisfying services and storage capacity requirements. The optimal system configuration is obtained by solving the above model using the generalized Lagrange multiplier method. Ashayeri et al. (1985) solve a similar problem as Karasawa et al. (1980). Given the storage capacity requirement and the height of racks, their models can be simplified to include only a single design variable, i.e., the number of aisles. Furthermore, the objective function is shown to be convex in the number of aisles, which allows a simple one-dimensional search algorithm to optimally solve the problem. Rosenblatt et al. (1993) propose an optimization model that is a slight modification of Ashayeri et al. (1985), which allows a crane to serve multiple aisles. A combined optimization and simulation approach is proposed. The optimization model is solved to obtain an initial design, which might not satisfy some performance constraints (e.g., for service level) that are difficult to model analytically. These performance measures are then evaluated in a simulation using the outputs from the optimization model. If the corresponding constraints are satisfied, then the procedure stops. Otherwise, the optimization model is altered by adding new constraints (which are constructed by approximating the simulation results) and solved again to find another design. It is reported that the optimal solution can be found in a few iterations. Zollinger (1996) and Malmborg (2001) proposes some rule of thumb heuristics in designing an AS/RS. The design criteria include the total equipment costs, S/R machine utilization, service time, number of jobs waiting in the queue, and storage space requirements. Some other less well-discussed AS/RS design problems include determining the size of the basic material handling unit and the configuration of I/O points. Roll et al. (1989) propose a procedure to determine the optimal size of containers in an AS/RS, which is the basic unit for storage and order picking. Assuming only one container size is used, container size has an direct effect on space utilization, and therefore on the equipment cost given the storage capacity requirement needs to be satisfied. The proposed approach determines the optimal container size to minimize the relevant equipment cost. Randhawa et al. (1991) and Randhawa and Shroff (1995) use simulations to investigate different I/O configurations on performance such as throughput, mean waiting time, and maximum waiting time. The results indicate that increased system throughput can be achieved using different I/O configurations instead of the common one-dock layout where the dock is located at the end of the aisle.

3.2.4 Equipment selection

The equipment selection problem is to determine the level of automation in a warehouse, and decide what type of storage and material handling systems should be employed. This decision obviously is a strategic one that affects almost all the other decisions, and constrains the overall warehouse investment and performance. Selecting a suitable level of automation is far from obvious, and in practice it is usually determined based on the personal experience of designers and managers. Academic research in this category is extremely rare. Cox (1986) provides a methodology to evaluate different levels of automation based on a cost-productivity analysis technique called the hierarchy of productivity ratios. White et al. (1981) develop analytical models to compare block stacking, single-deep and double-deep pallet rack, deep lane storage, and unit load AS/RS in order to determine the minimum space design. Matson and White (1981) extend White et al. (1981) to develop a total cost model incorporating both space and material handling costs, and demonstrate the effect of handling requirements on the optimum storage design. Sharp et al. (1994) compare several competing small part storage equipment types assuming different product sizes and dimensions. They considered shelving systems, modular drawers, gravity flow racks, carousel systems, and mini-load storage/retrieval systems. The costs they considered include operational costs, floor space costs, and equipment costs. In summary, research on equipment selection is quite limited and preliminary, although it is very important in the sense that it will affect the whole warehouse design and the overall lifetime costs.

3.2.5 Operation strategy

This section discusses the selection of operation strategies in a warehouse. The focus is given to operation strategies that, once selected, has important effects on the overall system and is not likely to be changed frequently (e.g., use of randomized storage or dedicated storage, or use zone picking or not). Chapter 2 discusses in detail different operation policies and their implementations for receiving, storage, order picking, and shipping. This section will discuss the literature on the comparison of operational strategies, which provides some guides as to which operational strategies should be selected in a warehouse. Two major operation strategies are discussed, i.e., the storage strategy and the order picking strategy.

The basic storage strategies include random storage, dedicated storage, class-based storage, and DOS-based storage, as explained in chapter 2. Hausman et al. (1976), Graves et al. (1977), and Schwarz et al. (1978) compare random storage, dedicated storage, and class-based storage in single-command and dual-command AS/RS using both analytical models and simulations. They show that significant reductions in travel time are obtainable from dedicated storage compared with random storage, and also that class-based storage with relatively few classes yields travel time reductions that are close to those obtained by dedicated storage. Goetschalckx and Ratliff (1990) and Thonemann and Brandeau (1998) show theoretically that DOS-based storage policies is the most promising policy in terms of minimizing traveling costs. In practice, DOS-based policies are difficult to implement since it requires the tracking and management of each stored unit in the warehouse. Also the performance of DOS-based policies depends greatly on factors such as the skewness of demands, balance of input and output flows, inventory control policies, and the detailed implementations. In a study by Kulturel et al. (1999), class-based storage and DOS-based storage are compared using simulations, and the former is found to consistently outperform the latter. This conclusion may be reached because the DOS model rarely hold true in practice. Finally, the above results are all for unit-load AS/RS; studies on other storage systems are rarely reported. Malmborg and Al-Tassan (1998) develop analytic models to evaluate the performance of dedicated storage and randomized storage in less-than-unit-load warehouses, but no general conclusions comparable to the unit-load case are given.

There are a number of order picking strategies including, for example, single-order picking, batching with sort-while-pick, batching with sort-after-pick, sequential zone picking with single order, sequential zone picking with batching, concurrent zone picking without batching in the zones, and concurrent zone picking with batching in the zones. Furthermore, these different order picking strategies can be used with or without wave picking. The performance of an order picking strategy depends on the characteristics of orders and products, service requirements, as well as the configuration of the warehouse. Research on the selection of an order picking strategy is very scarce, which might be a result of the complexity of the problem itself. Lin and Lu (1999) compare single-order picking and batch zone picking for different types of orders, which are classified based on the order quantity and the number of ordered items. Petersen (2000) simulates five different order-picking policies: single-order picking, batch picking, sequential zone picking, concurrent zone picking, and wave picking. Two control variables in the simulation study are numbers of daily orders and demand skewness, while the other factors such as warehouse layout, storage assignment, and zone configuration (when zone and wave picking is used) are fixed. The performance measures used to compare the different policies include: the mean daily labor, the mean length of day, and the mean percentage of late orders. For each order picking policy, the simplest rules regarding batching, routing, and wave length are used. It should also be noted that the performance measures are mainly related to order picking efficiencies and service qualities; additional costs caused by downstream sortation using batch, zone, and wave picking are not considered. Furthermore, comparison of these policies are made mainly with regards to the order structures, while other important factors such as storage assignment and detailed implementations of the order picking policies are assumed to be fixed. Therefore, the results should not be considered as general and more research in this direction might be worthwhile to provide more guidance for warehouse designers.

3.3 Performance evaluation

Performance evaluation provides feedback on the quality of a proposed design and/or operational policy, and more importantly, on how to further improve it. There are different approaches for performance evaluation: benchmarking, analytic models, and simulations. This section will only discuss benchmarking and analytic models.

Warehouse benchmarking is a process of systematically assessing the performance of a warehouse, identifying inefficiencies, and proposing improvements. Data Envelopment Analysis (DEA) is regarded as an appropriate tool for this task because of its capability to capture simultaneously all the relevant inputs (resources) and outputs (performances), to construct the best performance frontier, and to reveals the relative shortcomings of inefficient warehouses. Schefczyk (1993), Hackman et al. (2001), and Ross and Droge (2002) shows some approaches and case studies of using DEA in warehouse benchmarking. An Internet-based DEA system (iDEAS) for warehouses is developed by the Keck Lab in Georgia Tech, which includes information of more than 200 warehouses (McGinnis (2003)).

Most of the literature on warehouse performance evaluation addresses analytic models for a specific performance measure, especially (or exclusively) for travel time estimation. Travel time models deal with the estimation of expected travel time per order picking tour given warehouse type, layout, and storage and order picking policy. They are classified as models for unit-load AS/RS, man-on-board AS/RS, end-of-aisle AS/RS, carousel and rotary racks, and conventional multi-aisle systems, as shown in Table 3.2. The factors that affect travel time including warehouse layout (for example, rack dimensions for AS/RS or number and length of aisles for conventional multi-aisle systems), storage location assignment policies, and routing policies. In general, the effects of routing policies are difficult to quantify analytically, which explains the relatively small number of papers for conventional multi-aisle systems that require the more complex routing of multiple locations. As a result, some basic routing policies are usually assumed to simplify the modeling, for example, the first-come-first-serve policy

(Graves et al. (1977)) or nearest-neighbor heuristic (Han et al. (1987)) for dual-command systems, the traversal policy (Hall (1993)) or return policy (Hall (1993) and Caron et al. (1998)) for conventional multi-aisle systems. The readers should keep this in mind when referring to the literature in Table 3.2.

The travel time models for AS/RS usually assume that one S/R machine serves one aisle, and the S/R machine travels at a constant speed ignoring acceleration/deceleration. Hwang and Ko (1988) develop travel time models for the case where multiple aisles can be served by a single S/R machine, and propose a procedure to find the minimum number of S/R machines and to identify the number of aisles each S/R machine serves. Hwang and Lee (1990) develop travel time models that consider the operating characteristics of the S/R machine such as the acceleration/deceleration rate and the maximum velocity. Chang and Wen (1997) and Chang et al. (1995) consider a similar problem where the S/R machine has various travel speeds and known acceleration/deceleration rates, and use the travel time models to determine the optimal rack configuration.

Table 3.2 Literature of travel time models for different warehouse systems

		Randomized Storage	**Dedicated Storage**	**Class-based Storage**
Unit-Load AS/RS	**Single-Command**	Hausman et al. (1976) Bozer and White (1984) Thonemann and Brandeau (1998) Kim and Seidmann (1990) Hwang and Ko (1988) Lee (1997) Hwang and Lee (1990) Chang et al. (1995) Chang and Wen (1997) Koh et al. (2002) Lee et al. (1999)	Hausman et al. (1976) Thonemann and Brandeau (1998) Kim and Seidmann (1990)	Hausman et al. (1976) Thonemann and Brandeau (1998) Rosenblatt and Eynan (1989) Eynan and Rosenblatt (1994) Kouvelis and Papanicolaou (1995) Kim and Seidmann (1990) Pan and Wang (1996) Ashayeri et al. (2002)
	Dual-Command	Graves et al. (1977) Bozer and White (1984) Kim and Seidmann (1990) Hwang and Ko (1988) Lee (1997) Han et al. (1987) Hwang and Lee (1990) Chang et al. (1995) Chang and Wen (1997) Koh et al. (2002) Lee et al. (1999)	Graves et al. (1977) Kim and Seidmann (1990)	Graves et al. (1977) Kouvelis and Papanicolaou (1995) Kim and Seidmann (1990) Pan and Wang (1996) Ashayeri et al. (2002)
	Multi-Shuttle	Meller and Mungwattana (1997)		
Man-on-Board AS/RS		Hwang and Song (1993)		
End-of-Aisle AS/RS		Bozer and White (1990) Bozer and White (1996) Foley and Frazelle (1991) Park et al. (1999)	Park et al. (2003)	
Carousel and Rotary Racks		Han and McGinnis (1986) Han et al. (1988) Hwang et al. (1999)		
Conventional Multi-aisle System		Hall (1993) Jarvis and McDowell (1991) Chew and Tang (1999)	Caron et al. (1998) Caron et al. (2000) Jarvis and McDowell (1991) Chew and Tang (1999)	Jarvis and McDowell (1991) Chew and Tang (1999)

Other throughput related performance measures can be derived based on the travel time models, such as the total average service time (including waiting time and travel time), the average queue length, and the system throughput by using queuing models. In this case, the distribution of travel time instead of just the average is usually required to form the queuing models. Foley and Frazelle (1991) develop the travel time distribution for AS/RS with randomized storage. If the exact distribution cannot be derived, it is usually approximated by a general distribution using its expected value and variance, e.g., Bozer and White (1984) for AS/RS with randomized storage, Park et al. (2003) for AS/RS with dedicated storage, and Chew and Tang (1999) for conventional multi-aisle systems. Furthermore, detailed information about the travel time distribution is usually unavailable in the design phase due to uncertainties with the operational policies. Therefore, Foley et al. (2002) develop tight upper and lower bounds on throughput given only partial information about the travel time distribution.

Chow (1986) models the AS/RS as an M/G/1 queue in order to derive the average request waiting time and the average queue length. Lee (1997) also presents a stochastic analysis of the unit-load AS/RS using a single-server queuing model. Azadivar (1986) determines the throughput of a unit-load AS/RS using a stochastic constrained optimization problem, where the constraints are on the maximum storage queue length and the average waiting time for retrieval requests. Malmborg (2000) evaluates performance measures such as S/R machine utilization, queue lengths, average cycle time, and expected waiting time for a twin shuttle AS/RS. Bozer and White (1990) consider end-of-aisle order picking systems with random storage, and use the approximated travel time distribution discussed in the last paragraph to derive the system throughput. Bozer and White (1996) extend Bozer and White (1990) to more general end-of-aisle order picking systems, which might have multiple pick positions per aisle and multiple aisles per picker. Park et al. (2003) determines the throughput of end-of-aisle order picking systems with turnover-based storage. Park et al. (1999) further investigate the effects of buffer sizes on the throughput of end-of-aisle order picking systems using a two-stage cyclic queuing model. While the above research has been focused on unit-load AS/RS, Chew and Tang (1999) develop a travel time model for conventional multi-aisle warehouses with general storage assignment, which gives the exact probability mass functions as well as the first and second moments that characterize the order picking tour length. They then apply the model to analyze order batching and storage allocation by a queuing model. Bhaskaran and Malmborg (1989) also present a stochastic performance evaluation model on the service process for multi-aisle warehouses with an approximated distribution for the service time that depends on the batch size and the travel distance. de Koster (1994) develops queuing models to evaluate the performance of a warehouse that uses sequential zone picking where each bin are assigned to one or more orders, and are transported using a conveyer. If a bin needs to be picked at a specific zone, it is transported to the corresponding pick station. After it is picked, it is then put on the conveyor to be sent to the next pick station. The proposed queuing network model evaluates performance measures such as system throughput, picker utilization, and the average number of bins in the system based on factors such as the speed and length of the conveyor, the number of picking stations, and the number of picks per station.

The above analytical performance evaluation models have been concentrated on throughput-related performance, especially, on the travel time analysis and the service quality in processing storage and retrieval requests. Other performance measures might also be very important for a warehouse, e.g., storage capacity, construction cost, and operational cost, for which few sophisticated analytical approaches are available. Furthermore, it is important to have integrated models that can evaluate the tradeoffs between different performance measures in a unified way. Such integrated models are especially useful in the early design phase. However, research results in this direction are limited. Malmborg (1996) proposes an integrated performance evaluation model for a warehouse that has a forward-reserve configuration. The proposed model evaluates costs associated with: storage capacity; space shortage; inventory carrying, replenishing, and expediting; order picking; and internal replenishment for the forward area, based on information about inventory management, forward-reserve space allocation, and storage layout. Malmborg and Al-Tassan (2000) presents a mathematical model to estimated space requirements and order picking cycle times for less than unit load order picking systems that uses randomized storage. The inputs of the model include product parameters, equipment specifications, operational policies, and storage area configurations. Malmborg (2003) models the dependency of performance measures such as expected total system construction cost and throughput on factors such as the vehicle fleet size, the number of lifts, and the storage rack configurations for warehouse systems that use rail guided vehicles.

3.4 Case studies

Various warehouse design and operation problems have been discussed in this and the previous chapter. This section lists some real industrial case studies, which not only provide applications of the various design and operation methods in practical contexts, but more importantly also identify possible future research challenges from the industrial point of view. Table 3.3 lists these case studies with the problems and the types of warehouse they investigated. The detailed results and discussions are too cumbersome to be presented here. Interested readers should refer to the original papers. In general, these case studies demonstrate that substantial benefits might be achieved by appropriately designing and operating a warehouse, see for example Zeng et al. (2002) and van Oudheusden et al. (1988). On the other hand, many practical complications might arise when applying even the simplest rule in a practical context, for example, the COI-based storage location assignment rule (Kallina and Lynn (1976)). Some of these complications have been addressed in the academic research, but many others are still remained unexplored. These and more industrial case studies will help the warehouse research community to better understand the real issues and to make a more substantial impact on the practice.

Table 3.3 A Summary of the literature on warehouse case studies

Citation	Problems studied	Type of warehouse
Cormier and Kersey (1995)	Conceptual design	A warehouse for perishable goods that requires Just-In-Time operations
Yoon and Sharp (1995)	Conceptual design	An order picking system
Zeng et al. (2002)	Storage location assignment; warehouse dimensioning; storage and order picking policy	A distribution center
Kallina and Lynn (1976)	Storage location assignment using the COI rule	A distribution center
Brynzer and Johansson (1995)	Process flow; batching; zone picking;	Kitting systems that supply materials to assembly lines
Burkard et al. (1995)	Vehicle routing	An AS/RS where a S/R machine can serve any aisle using a switching gangway
van Oudheusden et al. (1988)	Storage location assignment; batching; routing	A man-on-board AS/RS in an integrated steel mill
Luxhoj and Skarpness (1986)	Manpower planning	A distribution center
Johnson and Lofgren (1994)	Simulation by decomposition	A distribution center

3.5 Computational systems

This section describes some computational tools that have been developed to aid in the design and operation of a warehouse, i.e., Computer-Aided Warehouse Design and Planning systems (CAWD and CAWP). There are numerous commercial Warehouse Management Systems (WMS) available in the market, which basically help the warehouse manager to keep track of the product, order, space, equipment, and human resource in a warehouse, and provide rules/algorithms for storage location assignment, order batching, pick routing, etc. Detailed review of these systems is beyond the scope of this chapter. Instead, we focus on the discussion of some prototyping systems developed by academic researchers. As previous sections show, research on various warehouse design and operation problems has being going on for almost half a century, and as a result, a large number of methodologies, algorithms, and empirical studies have been generated. However, we haven't seen many successful implementations of these academic results in current commercial WMS systems. The prototyping systems discussed in this section might shed some lights on how academic research results could be utilized to develop more sophisticated computer aided warehouse design and operation systems.

Perlmann and Bailey (1988) presents a computer-aided design software that allows a warehouse designer to quickly generate a set of conceptual design alternatives including building shape, equipment selection, and operational policy selection, and to select among them the best one based on the specified design requirements.

Luxhoj et al. (1993) develop an expert system to select inventory control policies based on information on, for example, demand, lead-time, and suppliers. Different inventory control models are linked with the expert system to calculate detailed operational parameters, such as order quantity and safety stock level, once an inventory policy is selected. Linn and Wysk (1990) develop an expert system for the control of an AS/RS in a dynamic environment. A control policy determines decisions such as storage location assignment, which item to retrieve if multi-items for the same product are stored, storage and retrieval sequencing, and storage relocation. Several control rules are available for each decision and the control policy is constructed by selecting one individual rule for each decision in a coherent way based on the dynamically changing system states such as demand pattern and traffic intensity. A similar AS/RS control system is proposed by Wang and Yih (1997) based on neural networks.

Ito et al. (2002) propose an intelligent agent based system to model a warehouse, which is composed of three subsystems, i.e., agent-based communication system, agent-based material handling system, and agent-based inventory planning and control system. Seven basic agents are developed including customer, supplier, order, inventory, product, supplier-order, and automatic-guided vehicle, which communicate with each other within the framework of the system. The proposed agent-based system is used for the design and implementation of warehouse simulation models. Kim et al. (2002) presents an agent based system for the control of a novel warehouse for cosmetic products. Besides the communication function, the agents also make decisions regarding the operation of the warehouse entities they represented in a dynamic real-time fashion. Since the decision made by an agent affects other agents, a proper coordination scheme among agents in the system needs to be developed. The authors propose a hybrid framework for the coordination of agents, which combines the advantages of both hierarchical and heterarchical schemes to allow coordination between different levels as well as within the same level.

3.6 Conclusions and discussions

Figure 3.1 shows the distribution of the surveyed literature among the warehouse design problems. The numbers in parentheses represent the number of papers related to the corresponding problems. The total number of papers on warehouse design problems is 46, which is about 1/3 of the number of papers on warehouse operation problems. Although the number might not be exact, it reflects the general situation that most warehouse research efforts have been devoted to operation problems instead of design problems. This is not because warehouse design is less important than warehouse operation, but because warehouse design problems are much more difficult to treat analytically. The difficulties are: first, the design decisions are closely interrelated such that good decisions are only achievable by considering the decisions in an integrated way, but models integrating all design decisions are much more difficult to develop and analysis; second, the design problem has significant implications for warehouse operations, but in the design stage, it is usually not very clear how the warehouse is going to be operated; this introduces uncertainty in modeling the influence of design decisions on operation performances. These challenges need to be addressed in the future warehouse research.

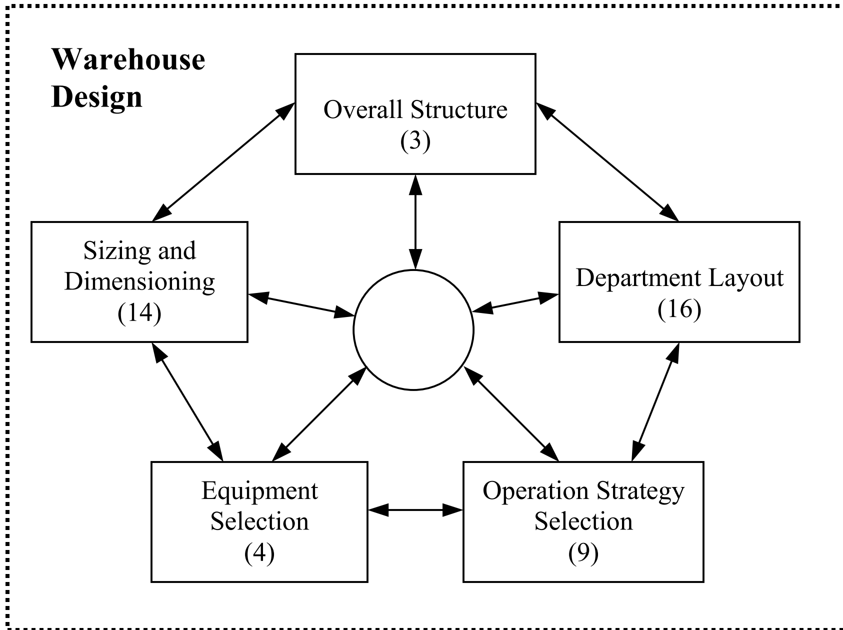


Figure 3.1 Illustration of the distribution of warehouse design literature

Warehouse performance evaluation has been an important topic in the past, but most of the proposed models focus on individual performance measures, such as travel time. Integrated models assessing overall warehouse performances are rare. Such integrated models are important in order to balance the tradeoffs among different performance criteria, and therefore deserve more attention in the future.

Most of the academic research results are not well validated and accepted in the industry. Research on developing industrial case studies and computer aided warehouse design and operation tools is very limited. More practical case studies might help us to realize the potential benefits of applying academic research results to real problems, and identify the hidden challenges that prevent their successful implementations. On the other hand, more sophisticated computer aided design and operation systems can facilitate the use of advanced methods in practice by imbedding them in the computational systems such that the warehouse designers and/or managers can utilize them without bothering too much about implementation details. The impact of computational systems on practice has been successfully demonstrated in other engineering disciplines, such as, computer aided design tools widely and effectively used in almost every mechanical or electrical design project today, resulting in better design results in a shorter cycle time. Similar impacts are not yet seen in the warehouse design and operation area, and future research efforts might contribute to develop more sophisticated computer aided design and operation systems to greatly improve the current warehouse practice.

Finally, both analytic and simulation models are currently used to analyze warehouse problems. Both methods have their advantages and disadvantages. Analytic models are usually design-oriented in the sense that they can explore many alternatives quickly to find the optimal (or near-optimal) solution. But it is usually difficult to develop analytic models that can capture all the relevant details of the system. When such models do exist, they are usually too difficult to solve in practice. On the other hand, simulation models are usually analysis-oriented in the sense that given a set of design and operational parameters, detailed performance measures can be obtained by building and running a simulation model. But their capability to explore a large number of alternatives is limited. It seems that there is a need to integrate both approaches to achieve more flexibility in analyzing warehouse problems. This is also pointed out by Ashayeri and Gelders (1985), and its applicability has been demonstrated by Rosenblatt and Roll (1984) and Rosenblatt et al. (1993).

CHAPTER 4

THE SIZING AND DIMENSIONING OF A FORWARD-RESERVE WAREHOUSE

4.1 Introduction

Storage and order processing are two basic functions of a warehouse and they have different and often conflicting requirements. For example, the use of high-density storage technologies, such as block stacking or deep-lane pallet racks, maximizes the space utilization; however, these technologies are inefficient for order picking since the goods are not easily accessible. Order picking benefits if goods are stored in a compact area with sufficient aisle space not only for convenient item access but also with limited non-productive walk time between order picking stops. Such an arrangement may not provide enough storage capacity for required inventory so that a secondary storage area may be required for the excess.

The forward-reserve configuration is a popular warehouse design strategy that facilitates efficient order picking while maintaining sufficient storage capacity. The primary function of the forward area is order picking. It is compact in size and uses equipment types such as bin shelving and gravity flow rack to allow convenient item selection and retrieval. The primary function of the reserve area is storage, where goods are stored in media such as block-stacked pallets or pallet racks to achieve high space utilization. The fundamental characteristic of the forward-reserve configuration is the dedication of different warehouse areas to different warehouse functions, i.e., order picking in the forward area and storage in the reserve area, so that their respective advantages can be fully utilized and warehouse construction and operation cost can be minimized.

A number of papers have discussed the forward-reserve warehouse. Their main focus has been on the tactical level, i.e., the forward-reserve allocation problem, which assumes the forward area has a given limited size and determines which SKUs should be assigned to the forward area and in what quantity to minimize the total order picking and internal replenishment cost (Hackman and Rosenblatt (1990)). This chapter focuses on the sizing and dimensioning of a forward-reserve warehouse, i.e., the problem of determining warehouse dimensions and allocating space between the forward and reserve areas to minimize the total warehouse life cycle cost. Compared with the forward-reserve allocation problem, the sizing and dimensioning problem is a strategic level design decision. The costs affected by the sizing and dimensioning decisions are construction cost, inventory cost, and material handling cost; these costs need to be carefully balanced in order to minimize the total life cycle cost.

We propose a mathematical model for the forward-reserve sizing and dimensioning problem, and develop an optimal solution algorithm based on Generalized Benders Decomposition (GBD). Section 4.2 presents the mathematical model for the forward-reserve sizing and dimensioning problem. Section 4.3 develops the solution algorithm starting with a brief description of GBD. Section 4.4 presents the computational results. Finally, conclusions and future research directions are given in Section 4.5.

4.2 Mathematical models

In developing our model, we assume that pallet racks and shelves are used in the reserve and forward area respectively, and the basic block layout of the forward-reserve warehouse is as illustrated in Figure 4.1. Some variations of this basic block layout are discussed in section 4.3.3.

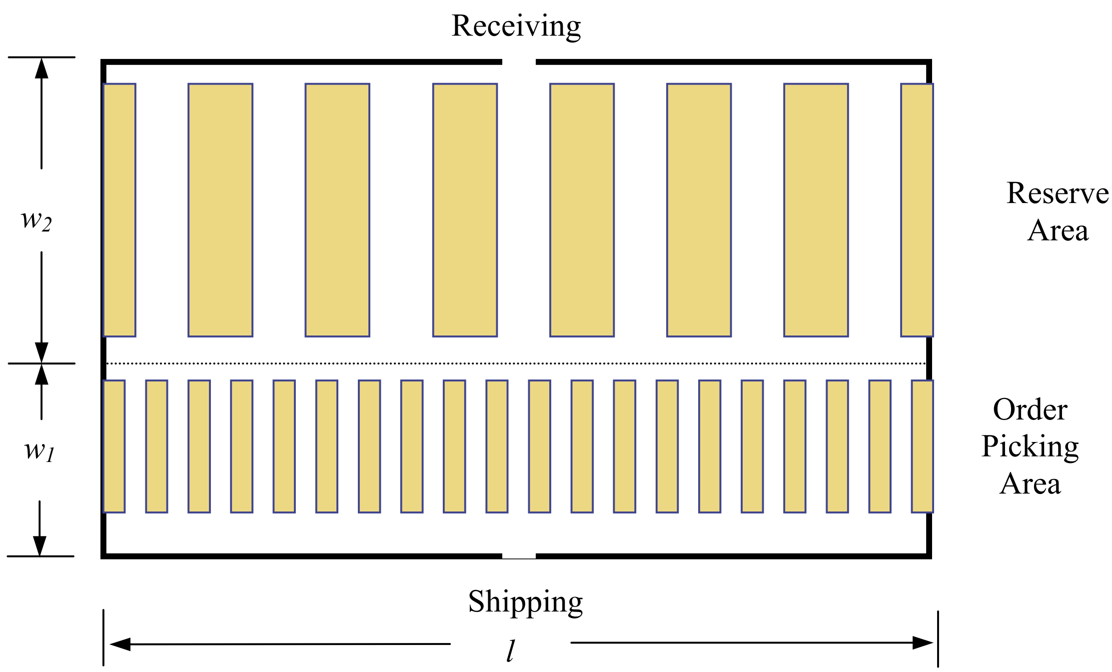


Figure 4.1 A block layout of the forward-reserve warehouse

The following notation will be used throughout this chapter:

Parameters:

A_b – sum of the travel aisle width in the reserve area and twice the depth of a pallet rack

A_f – sum of the travel aisle width in the forward area and twice the depth of a shelf

A_s – width of a shelf (measured along the travel aisle)

A_p – width of a pallet (measured along the travel aisle)

A_i – fixed ordering costs for SKU i per external replenishment

C_{a1} – construction cost per area unit of the forward area

C_{a2} – construction cost per area unit of the reserve area

C_w – construction cost per length unit of the external walls

C_i – inventory holding cost per volume unit (e.g. cubic meter) per year for SKU *i*

C_o – order picking cost per unit of travel distance

C_p – put-away cost per unit of travel distance

C_r – internal replenishment cost per unit of travel distance

D_i – annual demand of SKU *i* expressed in volume units (e.g., cubic meters)

I – index set of the SKUs

N – number of the SKUs

N_p – average number of picks per order picking trip

N_r – average number of order picking trips per year

V_p – net volume of product stored in a pallet

β_b – space utilization factor for the reserve area, defined as the net volume of product stored per unit area

β_f – space utilization factor for the forward area

T – planning time horizon of the warehouse measured in years

r – discount rate

S_i – safety stock level of SKU *i*

Z – net present value discount factor

$$Z = \sum_{t=1}^{T} (1+r)^{-t}$$

Variables:

y_l – number of aisles in the forward area.

y_{w1} – number of shelves per aisle in the forward area

y_{w2} – number of pallet positions per aisle in the reserve area

q_i – external order quantity for SKU i in volume units, e.g., cubic meters

z_i – quantity of SKU i allocated to the forward area in volume units, e.g., cubic meters

Assumptions:

- Demand rate is constant over the planned time horizon.
- The internal replenishment is assumed to be instantaneous so that a SKU is replenished when its inventory in the forward area reaches zero. An external replenishment happens when the total inventory in the warehouse drops to a given safety stock level based on the SKUs' demand rate and lead-time for replenishment.
- Randomized storage is used in both reserve and forward areas.
- Orders are batch picked from the forward area.
- The internal replenishment for any SKU can be performed in a single trip, while the put-away from receiving to the reserve area is performed one pallet a time.
- The clear height of the warehouse is given.

The warehouse dimensions are determined by:

$$l = y_l A_f; \quad w_1 = y_{w1} A_s; \quad w_2 = y_{w2} A_p$$

Note that the width of cross aisles, which can be added as a constant to the above formula, is not included in the calculation of dimensions to simplify the notation.

The warehouse construction cost is modeled as a function of the warehouse area and perimeter, following White and Francis (1971) and Bassan et al. (1980):

$$ConstructionCost = C_{a1} l w_1 + C_{a2} l w_2 + 2C_w (l + w_1 + w_2) \tag{4.1}$$

The material handling cost includes the cost for put-away, internal replenishment, and order picking. A put-away trip starts at the receiving door, goes to a location in the reserve area, stores the pallet, and then returns to the receiving door to store another pallet. Assuming randomized storage, the average travel distance per put-away trip can be calculated based on a continuous approximation of the storage area as follows:

$$\frac{2}{lw_2} \int_{-l/2}^{l/2} \int_0^{w_2} \left(|x_1| + |x_2|\right) dx_1 dx_2$$

The rectilinear distance metric is used since the trip is single-command and follows the aisle structure. Integrating this equation and multiplying the result by the cost coefficient, we obtain the average put-away cost per trip as:

$$c_p = C_p (\frac{l}{2} + w_2) \tag{4.2}$$

An internal replenishment trip starts by picking the required items from the reserve area, traveling to the forward area to place the items, and then returning to the reserve area to start the next task. The travel distance per internal replenishment is calculated by:

$$\frac{1}{l^2 w_1 w_2} \int_{w_2}^{w_1 + w_2} \int_{-l/2}^{l/2} \int_0^{w_2} \int_{-l/2}^{l/2} 2(| x_1 - x_2 | + | y_1 - y_2 |) dx_1 dy_1 dx_2 dy_2$$

Therefore, the average internal replenishment cost per trip can be calculated as:

$$c_r = C_r (\frac{2}{3} l + w_1 + w_2) \tag{4.3}$$

Assuming randomized storage and a traversal routing policy, the average order picking cost per batch picking tour can be modeled following Hall (1993) as shown in (4.4). Travel cost models for other storage and routing policy can be found in the literature listed in Table 3.2.

$$C_o\left(2l\left(\frac{N_p-1}{N_p+1}\right)+w_1 y_l\left(1-\left(\frac{y_l-1}{y_l}\right)^{N_p}\right)+0.5w_1\right) \tag{4.4}$$

Therefore, the total annual material handling cost is:

$$\begin{aligned} MHCost &= \sum_i\left(c_r\frac{D_i}{z_i}+c_p\frac{D_i}{V_p}\right) \\ &+N_r C_o\left(2l\left(\frac{N_p-1}{N_p+1}\right)+w_1 y_l\left(1-\left(\frac{y_l-1}{y_l}\right)^{N_p}\right)+0.5w_1\right) \end{aligned} \tag{4.5}$$

The above equation does not include the insertion and extraction cost of storing and picking SKUs from racks and/or shelves, which do not depend on the warehouse dimensions and can be modeled as constants. The external replenishment for a SKU is performed when the total inventory in the warehouse drops to its safety stock level S_i. Therefore, the total annual inventory holding and external replenishment cost is represented by:

$$InventoryCost=\sum_i(A_i\frac{D_i}{q_i}+C_i(\frac{q_i}{2}+S_i)) \tag{4.6}$$

The forward and reserve areas constitute a two-echelon inventory system. Figure 4.2 illustrates the inventory levels for SKU *i* in the warehouse where the dashed line represents the total warehouse inventory, the solid line represents the inventory in the reserve area, and t_0 and t_1 represent the times that external replenishments are performed. The maximum inventory for SKU *i* in the reserve area occurs when an external replenishment is performed, but its value is difficult to determine. For example, at time t_0, the inventory in the reserve area is $q_i + S_i - z_i$. But at time t_1, the inventory in the reserve area is close to $q_i + S_i$. Therefore, the maximum inventory of a specific SKU depends not only on its lot sizes and safety stock levels, but also on the timing of external and internal replenishments. Furthermore, the total required storage space is less than the sum of the maximum inventory for all SKUs since random storage is used and it is not likely that all SKUs achieve their maximum inventory level at the same time.

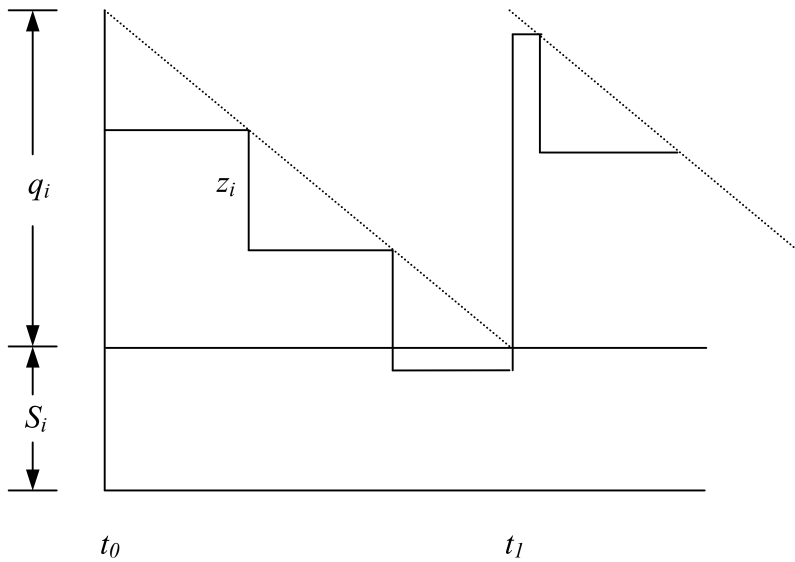


Figure 4.2 Inventory level in the warehouse

The effect of timing on the total required storage space is called staggering, which has been extensively studied in the literature (see, for example, Gallego et al. (1996)). The staggering effect can be represented as the following:

$$Staggering\ Factor = \frac{Total\ Storage\ Space\ Demand}{Sum\ of\ \ Maximum\ Storage\ Levels\ of\ \ Individual\ SKUs}$$

The sum of maximum storage levels of individual SKUs in the above formula is the total storage space requirement in the worst case, i.e., all SKUs achieve their maximum inventory level at the same time. The actual total storage space demand is always less than or equal to that in the worst case, therefore the staggering factor has a value between 0 and 1.

The warehouse design model needs to ensure that the warehouse can provide enough storage space in both the reserve and forward areas, which can be represented as follows:

$$Total\ Storage\ Space\ Demand \leq Net\ Available\ Storage\ Space$$

Or equivalently:

$$Sum\ of\ Maximum\ Storage\ Levels\ of\ Individual\ SKUs \leq \frac{1}{Staggering\ Factor} \times Net\ Available\ Storage\ Space$$

Therefore, the following inequalities represent the storage space constraints in the reserve and forward area respectively:

$$\sum_i (q_i + S_i) \leq \beta_b l w_2 \tag{4.7}$$

$$\sum_i z_i \leq \beta_f l w_1 \tag{4.8}$$

Factors β_b and β_b in (4.7) and (4.8) are compound factors which include not only the effect of staggering as discussed above, but also factors that determine the net available storage space for given warehouse dimensions, such as the warehouse clear height, volumetric limitations (e.g., obstructions in the rack area), and space utilization (e.g., the honeycombing phenomenon). The selection of values for β_b and β_b depend on the tradeoff between warehouse life-cycle cost and the risk of running out of storage space during peak inventory periods. A detailed discussion on these various factors and their typical values can be found in Section 3.2.1 of Sharp (2000). Furthermore, the maximum storage levels of SKU i in the reserve area is represented as $q_i + S_i$, which can be achieved (or nearly achieved) at certain specific time points but most of the time overestimates the actual storage space requirements as discussed earlier. This should be considered in selecting the value for β_b.

In summary, the forward-reserve warehouse sizing and dimensioning optimization model is:

$$\text{(P)} \qquad \min\ ConstructionCost + (InventoryCost + MHCost)Z$$

$$\text{s.t.} \quad (4.7) \text{ and } (4.8)$$

$$q_i, z_i \geq 0, \forall i \in I$$

$$y_l, y_{w1}, y_{w2} \in Z_+^3$$

The above model is based on the fluid assumption (Bartholdi and Hackman (2005)), which approximates each SKU as an incompressible and continuously divisible fluid. The objective function represents the total life-cycle cost including the construction cost and the net present value of the discounted operation cost. The model can also include other constraints, such as physical layout constraints and/or bounds on the space allocations, as discussed in Section 4.3.3. This problem is a non-convex mixed integer problem with a large number of continuous variables. In general, such problems are difficult to solve by general-purpose optimization packages. An efficient algorithm that exploits the special problem structure is developed in the next section.

4.3 Solution method

If the warehouse dimension variables (y_l, y_{w1}, y_{w2}) in problem (P) are temporarily fixed, the remaining problem is convex with only continuous variables, and as discussed below is easy to solve. This property suggests using a decomposition strategy in solving problem (P). In the remaining parts of this section, a solution method based on Generalized Benders Decomposition (GBD) (Geofferion (1972)) is proposed.

4.3.1 Generalized Benders Decomposition

The Generalized Benders Decomposition approach can be applied to problems that have the following general form:

$$\min f(x,y) \quad s.t. \quad G(x,y) \le 0, \quad x \in X,\ y \in Y \tag{4.9}$$

where x and y are vectors of decision variables and G is a m-vector of constraint functions. By fixing y, the problem reduces to the following sub-problem:

$$v(y) \equiv \quad \min_x f(x,y), \quad s.t. \quad G(x,y) \le 0, x \in X \tag{4.10}$$

Assuming problem (4.10) is convex for any fixed $y \in Y$, problem (4.9) can be reformulated to the following equivalent form based on the duality of convex problems (see Geofferion (1972) for the details):

$$\min_{y \in Y,\, y_0 \in R} y_o \tag{4.11.1}$$

$$s.t. \quad \inf_{x \in X} \left\{ f(x,y) + u^t G(x,y) \right\} \le y_0 \quad \forall u \ge 0 \tag{4.11.2}$$

$$\inf_{x \in X} \left\{ \lambda^t G(x,y) \right\} \le 0 \quad \forall \lambda \in \Lambda \tag{4.11.3}$$

where:

$$\Lambda \equiv \left\{ \lambda \in R^m : \lambda \ge 0\, and \sum_{i=1}^{m} \lambda_i = 1 \right\}$$

Problem (4.11) is called the master problem, where the objective function (4.11.1) and constraints (4.11.2) enforce y_o equal to $v(y)$ (i.e., the optimal function value of the sub-problem) by duality and therefore the problem is to find an optimal y that minimizes $v(y)$; constraints (4.11.3) ensure the feasibility of a given y. There are an infinite number of constraints because μ and λ are continuous. The GBD algorithm employs a relaxation strategy to solve the master problem. The algorithm starts by solving a relaxed version of

(4.11) that includes only a subset of all the constraints, and the result ($\hat{y}$, $\hat{y}_o$) is passed to the sub-problem (4.10). Solving (4.10) for fixed ($\hat{y}$, $\hat{y}_o$) can result in three possible outcomes: (1) it is infeasible; (2) it is feasible, but $v(\hat{y}) > \hat{y}_o$; or (3) it is feasible, and $v(\hat{y}) \leq \hat{y}_o$. Note that $v(\hat{y})$ may be unbounded, which means the original problem is also unbounded. In case (3), the master problem is optimally solved. For case (1) or (2), the sub-problem will generate a value for the Lagrange multipliers λ or u such that the corresponding constraint will be violated by ($\hat{y}$, $\hat{y}_o$). The violated constraint is then added to the relaxed master problem and the process is repeated.

4.3.2 Solving the forward-reserve warehouse sizing and dimensioning model

Using the notation defined in Table 4.1, Problem (P) can be transformed to the following:

$$
\begin{aligned}
(\mathrm{P}') \qquad & \min \; f_1(y) + f_2(y,q,z) \\
& s.t. \; \sum_{i\in I} q_i \leq B_2 y_l y_{w2} + B \\
& \qquad \sum_{i\in I} z_i \leq B_1 y_l y_{w1} \\
& \qquad y \in Z_+^3,\; q_i, z_i \geq 0 \quad \forall\, i \in I
\end{aligned}
$$

where:

$$
\begin{aligned}
f_1(y) = {} & A_1 y_l + A_2 y_{w1} + A_3 y_{w2} + A_4 y_l y_{w1} + A_5 y_l y_{w2} \\
& + A_6 y_l y_{w1} \left(1 - \left(\frac{y_l - 1}{y_l} \right)^{N_p} \right) + A_7
\end{aligned} \tag{4.12}
$$

$$
f_2(y,q,z) = \sum_{i\in I} \left(\frac{\alpha_i}{q_i} + \beta_i q_i \right) + \sum_{i\in I} \left(\frac{\tau_i c_r}{z_i} \right) \tag{4.13}
$$

Note that f_2 depends on y since c_r is a function of the dimension variables as shown in (4.3). If y is fixed, (P’) reduces to the following two independent sub-problems:

$$
\text{(P1)} \quad \begin{aligned} \min \quad & \sum_{i \in I} \left(\frac{\alpha_i}{q_i} + \beta_i q_i \right) \\ s.t. \quad & \sum_{i \in I} q_i \le B_2 y_l y_{w2} + B \\ & q_i \ge 0 \quad \forall i \in I \end{aligned}
\qquad\qquad
\text{(P2)} \quad \begin{aligned} \min \quad & \sum_{i \in I} \left(\frac{\tau_i c_r}{z_i} \right) \\ s.t. \quad & \sum_{i \in I} z_i \le B_1 y_l y_{w1} \\ & z_i \ge 0 \quad \forall i \in I \end{aligned}
$$

Table 4.1 Notation for the definition of problem (P')

A_1	$2C_w A_f + 2C_o N_r A_f Z(N_p - 1)/(N_p + 1) + (\sum_{i \in I} D_i) C_p A_f Z/(2V_p)$
A_2	$2C_w A_s + 0.5 C_o N_r A_s Z$
A_3	$2C_w A_p + (\sum_{i \in I} D_i) C_p A_p Z / V_p$
A_4	$C_{a1} A_f A_s$
A_5	$C_{a2} A_f A_p$
A_6	$C_o N_r A_s Z$
A_7	$Z \sum_{i \in I} C_i S_i$
B	$-\sum_{i \in I} S_i$
B_1	$\beta_f A_f A_s$
B_2	$\beta_b A_f A_p$
α_i	$A_i D_i Z$
β_i	$C_i Z / 2$
τ_i	$D_i Z$

Problems (P1) and (P2) are both convex problems and can be solved very efficiently. The following describes the solution algorithms for solving (P1), (P2) and the master problem (P') respectively.

Solution algorithm for (P1)

Since (P1) is a convex problem, a feasible solution is optimal if it satisfies the Karush-Kuhn-Tucker (KKT) conditions as stated in equations (4.14):

$$\frac{\alpha_i}{q_i^2} - \beta_i - u = 0 \quad \forall i \in I \tag{4.14.1}$$

$$u(B_2 y_l y_{w2} + B - \sum_{i \in I} q_i) = 0 \tag{4.14.2}$$

$$u \geq 0 \tag{4.14.3}$$

The following algorithm can be used to find a feasible solution that satisfies the KKT condition and therefore solves (P1):

(1) Let $u = 0$, and

$$q_i = \sqrt{\frac{\alpha_i}{\beta_i}} \quad \forall i \in I \tag{4.15}$$

It is easy to check that the above result satisfies the KKT condition. If it is also feasible, i.e., it satisfies $\sum_{i \in I} q_i \leq B_2 y_l y_{w2} + B$, the algorithm stops with an optimal solution.

(2) Otherwise, we have $\sum_{i \in I} q_i > B_2 y_l y_{w2} + B$. Find a $u > 0$, and

$$q_i = \sqrt{\frac{\alpha_i}{\beta_i + u}} \quad \forall i \in I \tag{4.16}$$

such that $\sum_{i \in I} q_i = B_2 y_l y_{w2} + B$. The solution is feasible and satisfies the KKT condition, and therefore is optimal.

According to Equation (4.16), q_i is monotonically decreasing with u. Therefore, there is a unique solution of u at which $\sum_{i \in I} q_i = B_2 y_l y_{w2} + B$ is satisfied. A bisection search on u can be used in Step (2) to find this solution efficiently.

Solution algorithm for (P2)

Problem (P2) is even simpler since its objective function is a decreasing function in z_i, therefore the constraint $\sum_{i \in I} z_i \le B_1 y_l y_{w1}$ is always tight at the optimal solution. The KKT conditions for (P2) are:

$$\frac{c_r \tau_i}{z_i^2} - u = 0 \quad \forall i \in I \tag{4.17.1}$$

$$u(B_1 y_l y_{w1} - \sum_{i \in I} z_i) = 0 \tag{4.17.2}$$

$$u \ge 0 \tag{4.17.3}$$

From (4.17.1), we can calculate z_i as follows:

$$z_i = \sqrt{\frac{c_r \tau_i}{u}} \quad \forall i \in I \tag{4.18}$$

Since $\sum_{i \in I} z_i = B_1 y_l y_{w1}$ in the optimal solution, we can substitute (4.18) into it to get the optimal u:

$$u = \left(\frac{\sum_{i \in I} \sqrt{c_r \tau_i}}{B_1 y_l y_{w1}} \right)^2 \tag{4.19}$$

Therefore, the optimal solution for z_i can be calculated as follows:

$$z_i = \frac{\sqrt{c_r \tau_i}}{\sum_{i \in I} \sqrt{c_r \tau_i}} B_1 y_l y_{w1} \quad \forall i \in I$$

Solve the master problem (P')

The master problem takes the form of Problem (4.11). In our case, the sub-problems are always feasible for any $y \in Z_+^3$. Therefore, only constraints (4.11.2) need to be considered, which can be explicitly represented as:

$$\begin{aligned} & f_1(y) - u_1(B_2 y_l y_{w2} + B) - u_2 B_1 y_l y_{w1} \\ & \qquad + 2\sum_{i \in I} \sqrt{\alpha_i(\beta_i + u_1)} + 2\sum_{i \in I} \sqrt{\tau_i c_r u_2} \le y_o \end{aligned} \tag{4.20}$$

where u_1 and u_2 are the Lagrange multipliers of sub-problems (P1) and (P2).

Therefore, the master problem for the forward-reserve warehouse sizing and dimensioning problem can be stated as follows:

$$\min_{y \in Z_+^3, y_0 \in R} y_o \tag{4.21.1}$$

$$\begin{aligned} s.t. \quad & f_1(y) - u_1(B_2 y_l y_{w2} + B) - u_2 B_1 y_l y_{w1} + 2\sum_{i \in I} \sqrt{\alpha_i(\beta_i + u_1)} \\ & + 2\sum_{i \in I} \sqrt{\tau_i c_r u_2} \le y_o \quad \forall u_1, u_2 \ge 0 \end{aligned} \tag{4.21.2}$$

Problem (4.21) is a mixed integer nonlinear problem, but has only four variables, i.e., y_l, y_{w1}, y_{w2}, and y_o. Therefore, it can be solved using a Branch-and-Bound algorithm (for example, see Ryoo and Sahinidis (1996)).

Since u_1 and u_2 are continuous, problem (4.21) has an infinite set of constraints. It is solved with a relaxation strategy. The detailed algorithm for solving the master problem is as follows:

(1) Select a starting point $\bar{y}$, an initial upper bound UBD and a lower bound LBD, and the convergence tolerance parameter $\varepsilon > 0$.

(2) Solve the sub-problems (P1) and (P2) for $\bar{y}$. If the optimal objective value $v(\bar{y})$ of the sub-problem is less than the current upper bound, update UBD to $v(\bar{y})$. If LBD ≥ UBD - ε, terminate. Otherwise, determine the value of the multipliers $\bar{u}_1$ and $\bar{u}_2$, and add the corresponding constraint to the relaxed master problem.

(3) Solve the current relaxed master problem. Let $(\bar{y}, \bar{y}_0)$ be the optimal solution. $\bar{y}_0$ is the new lower bound, set LBD = $\bar{y}_0$. Go to step (2).

4.3.3 Variations of the model

Problem (P) can be extended to include other practical considerations, for example:

(1) Additional constraints $g(y) \le 0$ on the warehouse dimension variables can be added. These constraints may represent construction site limitations, construction budget limitations, and/or layout feasibility constraints. The same solution method developed in Section 4.3.2 still can be applied by directly adding these constraints to the master problem (4.21).

(2) Lower and upper bounds on the space allocation variables can be added to problem (P). They may represent bounds for a single space allocation variable in the form of $L_i \le q_i \le U_i$, or constraints on the total allocated space for a group of SKUs as represented by $L_k \le \sum_{i \in I_k} q_i \le U_k$, where $I_k \subset I$ is a subset of the SKUs. For illustration purpose, we will only discuss sub-problem (P1) by including the general upper bounds as follows:

$$
\begin{aligned}
\min \quad & \sum_{i \in I} \left(\frac{\alpha_i}{q_i} + \beta_i q_i \right) \\
s.t. \quad & \sum_{i \in I_k} q_i \le U_k \quad \forall k = 1, 2, \ldots, K \\
& q_i \ge 0 \qquad \forall i \in I
\end{aligned}
$$

The corresponding KKT conditions become:

$$\frac{\alpha_i}{q_i^2} - \beta_i - \sum_{\{k: q_i \in I_k\}} u_k = 0 \quad \forall i \in I$$

$$u_k (U_k - \sum_{i \in I_k} q_i) = 0 \quad \forall k = 1,2,\ldots,K$$

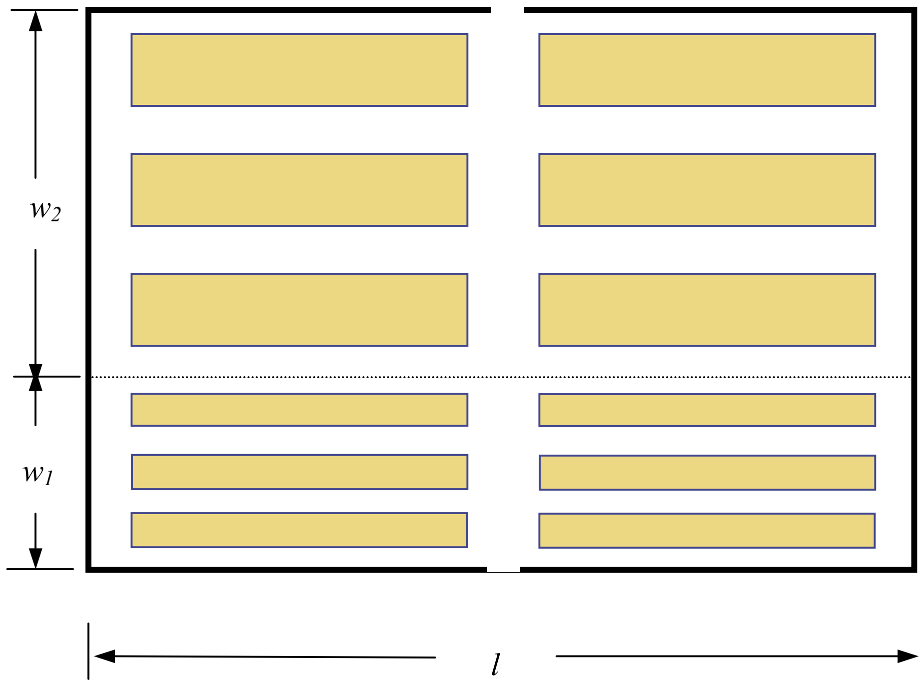
$$u_k \ge 0 \quad \forall k = 1, 2, \ldots, K$$

The modified solution algorithm is similar to that discussed in section 4.3.2. It starts by letting all $u_k = 0$ and $q_i = \sqrt{\alpha_i / \beta_i}$. If the solution is feasible, the algorithm stops with an optimal solution. Otherwise, the algorithm picks a u_k for which the corresponding constraint is violated (i.e., $\sum_{i \in I_k} q_i > U_k$), and $q_{i \in I_k}$ is decreased by increasing u_k until $\sum_{i \in I_k} q_i = U_k$. If the resulting solution is feasible, the algorithm stops; otherwise, the same process repeats until all constraints are satisfied.

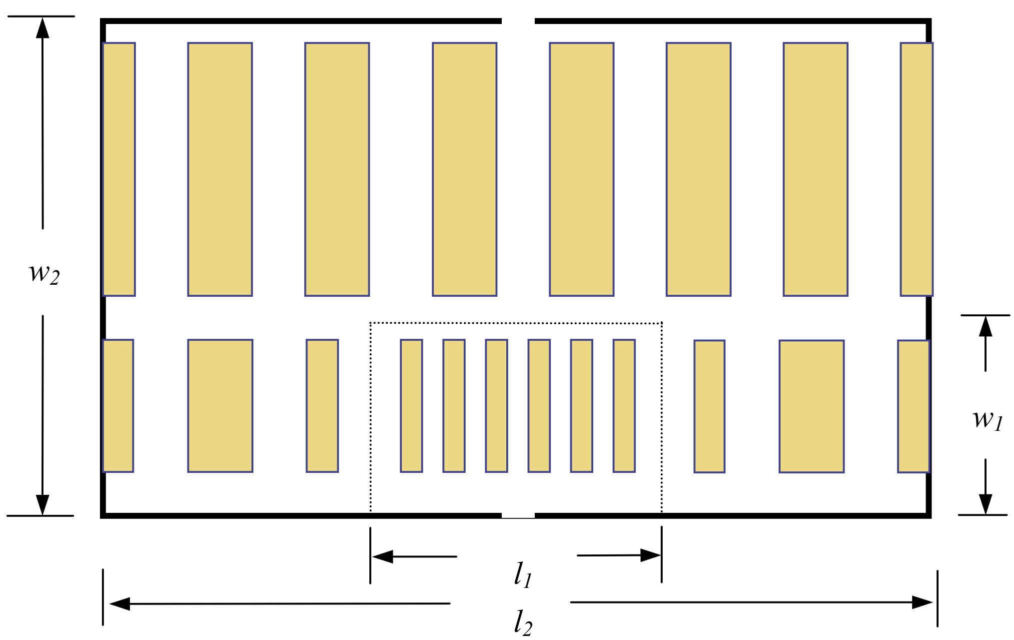
(3) It has been assumed that orders are only picked from the forward area. In reality, it might be beneficial to put only fast-moving SKUs in the forward area. In this case, orders will be picked from both the forward area for the fast-moving SKUs and from the reserve area for the slow-moving SKUs. If the set of fast-moving SKUs assigned to the forward area is given, then Equation (4.5) can be changed correspondingly to reflect the changes in order picking and

internal replenishment. The overall problem structure and solution method will remain the same. However, the GBD-based algorithm cannot be applied directly when the assignment of SKUs to the forward area needs to be determined by the model, since it requires introducing additional integer decision variables to represent the assignment and therefore results in a non-convex sub-problem. This problem will be discussed in chapter 6.

(4) The proposed approach can be extended to other types of forward-reserve block layouts, as shown in Figure 4.3. Figure 4.3(a) is similar to Figure 4.1 except for the orientation of the aisles. In this case, the major modification is to replace equation (4.4) with a different model to calculate the batched order picking cost for the horizontal aisle structure (for example, see Caron et al. (1998)). The block layout shown in Figure 4.3(b) has a U-shaped reserve area and its representation requires 4 dimension variables, i.e., l_1, w_1, l_2, w_2, with $l_1 < l_2$ and $w_1 < w_2$. The construction cost (i.e., equation (4.1)) and the average put-away and internal replenishment cost per trip (i.e., equation (4.2) and (4.3)) need to be revised following the same assumptions (e.g., a continuous approximation of the storage area and randomized storage).



(a) Horizontal aisle structure



(b) U-Shaped reserve area

Figure 4.3 Alternative block layouts of the forward-reserve warehouse

4.4 Numerical results

This section provides numerical evaluation of the proposed GBD-based algorithm. The algorithm is first compared with other solution methods to demonstrate its performance in terms of computational time. Sensitivity analysis is then performed to show how the uncertainty in design parameters affects the optimal solution.

The GBD-based algorithm is compared with two other solution methods, i.e., GAMS/CONOPT (Brooke et al. (1998)) and total enumeration. CONOPT solves Problem (P) as a continuous nonlinear problem, and therefore might result in a fractional solution. The final solution is obtained by rounding the fractional solution to its nearest integer solution. The enumeration method enumerates all the possible combinations of the dimension variables, solves the sub-problems for each of them, and finally selects the one that has the best objective value. Three problems are tested, which have different number of SKUs (8,000, 15,000 and 30,000 respectively) and represent different warehouse sizes. Table 4.2 shows the parameter values used to generate the testing problems. The GBD-based algorithm and CONOPT require the user to provide an initial design solution that might affect the computational time. In order to evaluate the effects of the starting point, three different starting points (i.e., y_{Init}^1, y_{Init}^2 and y_{Init}^3) are tested for each problem. Let ($y_l^*, y_{w1}^*, y_{w2}^*$) be the optimal solution of any of the three problems solved with the total enumeration method, y_{Init}^1, y_{Init}^2 and y_{Init}^3 are set as follows:

$$y_{Init}^1 = (y_l^* + 5, y_{w1}^* + 5, y_{w2}^* + 5)$$

$$y_{Init}^2 = (y_l^* + 15, y_{w1}^* + 15, y_{w2}^* + 15)$$

$$y_{Init}^3 = (y_l^* + 25, y_{w1}^* + 25, y_{w2}^* + 25)$$

Table 4.2 Parameter values for the numerical example

A_f	8
A_b	20
A_s	4
A_p	4
A_i	Uniform [20, 100]
C_{a_1}	30
C_{a_2}	25
C_w	1000
C_i	Uniform [0.3, 6.5]
C_o	0.0068
C_p	0.0024
C_r	0.0028
D_i	Uniform [10, 200]
N_p	80
V_p	64
β_b	5.38
β_f	1.75
T	10
r	0.2
Z	4.191

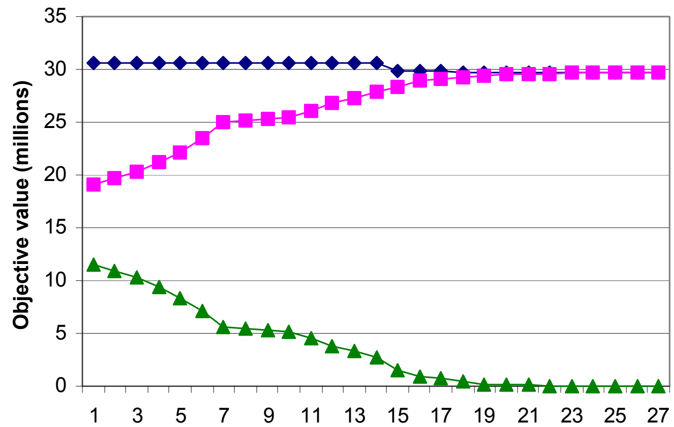
All tests are performed on a Sun 280R server with 2×900MHz UltraSparc-III CPU and 2GB RAM. Table 4.3 shows the warehouse size (i.e., $l*(w_1+w_2)$), shape ratio (i.e., $(w_1+w_2)/l$), and the ratio of the forward area and the total warehouse area (i.e., $w_1/(w_1+w_2)$) in the optimal solutions of the three tested problems. As the number of SKUs and the corresponding warehouse area increase, the warehouse shape ratio becomes smaller (with a diminishing decreasing rate), and the fraction of the warehouse area allocated to the forward area increases. It should be noted that these observations are based only on the single case investigated in the numerical experiment with all its listed assumptions. Table 4.4 shows the computational time for the different algorithms. The GBD-based algorithm is very efficient in solving the problem. In all the tested cases, it terminates with an optimal solution within 90 seconds. The efficiency of the GBD-based algorithm is not affected when the number of SKUs increases as compared with the other two algorithms. This is because the number of SKUs only affects the sub-problem, and the solution algorithm for the sub-problem is very efficient even when there are a large number of SKUs. Furthermore, the results in Table 4.4 also show that the solution time of the GBD-based algorithm is not sensitive to the starting points, which is a desirable property since the designer can be relieved from the effort to identify a good initial solution. Figure 4.4 illustrates the convergence history of the GBD-based algorithm for the problem with 15,000 SKUs using three different starting points. Although the initial gap is quite different, the number of iterations it takes for the algorithm to converge is similar for all three starting points; the behavior is similar for other problem sizes. Compared with the GBD-based algorithm, CONOPT can find a fractional solution that is close to the optimum and is efficient when the number of SKUs is small. However, the computational time of CONOPT increases dramatically as the problem size increases.

Table 4.3 Layout features of the optimal solutions

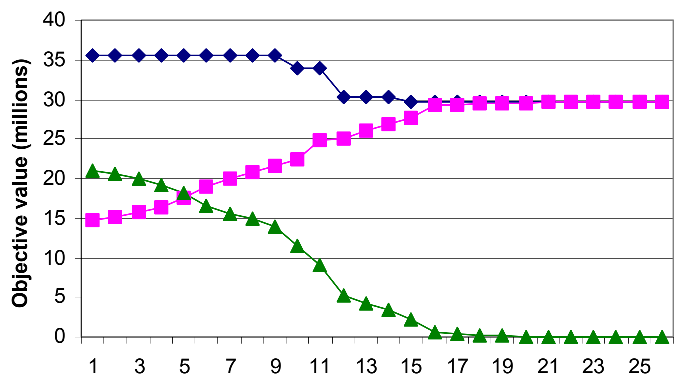
	$l*(w_1+w_2)$ (ft^2)	$(w_1+w_2) / l$	$w_1/(w_1+w_2)$
8000 SKUs	63232	0.68	0.15
15000 SKUs	128960	0.52	0.18
30000 SKUs	278880	0.4	0.23

Table 4.4 Computational time of different algorithms (seconds)

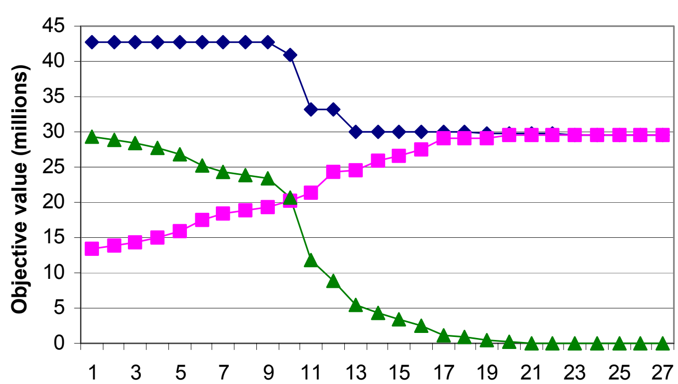
Number of SKUs	Starting Points	GBD	CONOPT	Enumeration
8000	y^1_{Init}	78.59	40.23	6358.83
	y^2_{Init}	73.27	47.96	
	y^3_{Init}	68.04	57.13	
15000	y^1_{Init}	81.78	246.29	14360.32
	y^2_{Init}	76.25	1047.28	
	y^3_{Init}	81.53	1437.58	
30000	y^1_{Init}	84.93	3614.84	26402.04
	y^2_{Init}	78.96	4045.73	
	y^3_{Init}	84.12	4516.99	



(a) Iterations with y^1_{init}



(b) Iterations with y^2_{init}



(c) Iterations with y^3_{init}

Upper bound Lower Bound Gap

Figure 4.4 Convergence of the GBD-based algorithm with 15,000 SKUs

Since many of the parameter values driving design are based on long-term forecasting, it is important to investigate how forecasting errors would affect the quality of the solutions. A set of parameters that may have a significant impact on the optimal solution is selected (i.e., $A_i, C_i, C_o, C_r, D_i, N_p$), and their values are varied to represent realizations that are different from the forecasts. Experiments are conducted on the three warehouse problems with 8000, 15,000, and 30,000 SKUs respectively. The values of the selected parameters are increased one at a time by 10%, 20%, 30%, and 40% relative to the forecasted values. By denoting p and p' as the forecasted and "realized" parameter values, $x(p)$ and $x(p')$ as the design based on the forecasted and "realized" values, $v(x(p))$ represents the total cost incurred for the forecasting-based design operated under the realized parameter values, and $v(x(p'))$ represents the optimal cost if the design had originally been based on the actual parameters values. The following formula measures the opportunity cost (or the cost of imperfect information) due to forecasting errors:

$$\frac{v(x(p)) - v(x(p'))}{v(x(p'))} \times 100\% \tag{4.22}$$

Numerical results summarized in Table 4.5 show that the design solution is quite robust with regard to the total warehouse life-cycle cost. In most cases, the loss is less than 1% of the cost for the ideal solution, i.e., assuming the realized parameter values were known perfectly at the design stage. The results also show that A_i (external replenishment cost) and D_i (demand rate) have a larger impact on the robustness of the solution from the cost perspective than the other parameters.

Table 4.5 Cost of imperfect information (%)

		8000 SKUs	15000 SKUs	30000 SKUs
Ai	10%	0.13	0.08	0.07
	20%	0.39	0.29	0.27
	30%	0.76	0.61	0.58
	40%	1.23	1.02	0.97
Ci	10%	0.00	0.01	0.01
	20%	0.02	0.05	0.06
	30%	0.06	0.11	0.12
	40%	0.11	0.18	0.20
Co	10%	0.01	0.01	0.01
	20%	0.06	0.08	0.08
	30%	0.15	0.19	0.16
	40%	0.26	0.32	0.29
Cr	10%	0.07	0.04	0.05
	20%	0.22	0.17	0.17
	30%	0.42	0.36	0.37
	40%	0.66	0.60	0.62
Di	10%	0.16	0.11	0.10
	20%	0.52	0.39	0.38
	30%	1.00	0.83	0.80
	40%	1.58	1.36	1.32
N_p	10%	0.00	0.00	0.00
	20%	0.01	0.02	0.03
	30%	0.03	0.04	0.06
	40%	0.05	0.08	0.11

A detailed look at the different cost elements reveals that the robustness is due to the fact that the proposed integrated model pools different costs so that tradeoffs among them can be balanced to minimize the effect of the change in design parameters on the total cost. Table 4.6 shows the change in the optimal warehouse dimensions and the cost of imperfect information (as decomposed into construction cost, inventory and external replenishment cost, and material handling cost) when the actual value of *Ai* is increased from the forecasted value by 20%, 40% and 200%. The optimal warehouse dimensions (y_l, y_{w1}, y_{w2}) based on forecasted values are (38, 8, 44). Intuitively, the optimal warehouse size will increase as *Ai* increases in order to provide more storage capacity to reduce the number of external replenishments. Therefore, the forecasting based design, as compared to the ideal design, will have a higher inventory and external replenishment cost (due to the increase in Ai), but a smaller construction and material handling cost (due to its smaller size). This results in a relatively small change in the total cost. However, this does not suggest that the warehouse size and dimensions can be determined arbitrarily. Although the total cost is robust when the value of design parameters varies in a small neighborhood of the forecasted value, the cost of imperfect information is significant when the design parameter is inappropriately estimated or the design has been determined arbitrarily by some ad-hoc methods (as illustrated in the case with a 200% change in Ai).

It should also be noted that in our experiments the design parameters are increased one at a time for the purpose of sensitivity analysis. Future research might also be performed to investigate the joint effects of different parameters on the quality of design solutions.

Table 4.6 Dimension change and cost of imperfect information (%) as Ai increases

Ai	y	Total Cost	Construction	Inventory & External Replenishment	Material Handling
20%	(40, 8, 47)	0.39	-8.22	5.48	-2.40
40%	(41, 8, 50)	1.23	-13.7	10.31	-4.32
200%	(49, 8, 64)	13.54	-37.49	44.59	-12.51

4.5 Conclusions

A GBD-based algorithm is proposed to solve the sizing and dimensioning problem for the forward-reserve warehouse to minimize the total life-cycle cost. Computational results demonstrate that it is very efficient in finding the global optimum, and the solution is quite robust with regards to uncertainty in design parameters. Chapter 6 will develop a model as well as a heuristic solution method that includes the decision of assigning SKUs to the forward area instead of assuming the assignment is given. This general model includes the forward reserve allocation problem as a sub-problem. We will focus first on this sub-problem in chapter 5.

CHAPTER 5

SOLVING THE FORWARD RESERVE ALLOCATION PROBLEM

5.1 Introduction

The previous chapter assumed that the forward area contains all SKUs so that order picking is performed only in the forward area. A major advantage of this arrangement is that it simplifies the order picking process by avoiding management complications involved in picking orders from different storage areas, such as order splitting and combination. However, one might also choose to assign only a subset of SKUs in the forward area, mainly for the following two reasons:

(1) The forward area usually has a limited storage capacity since it is compact in size and uses low-density storage equipment for efficient order picking. As more SKUs are assigned to the forward area, less space can be allocated to each SKU and consequently more frequent internal replenishing must occur.

(2) The picking activities are not evenly distributed among all SKUs. Some SKUs are fast movers for which demands occur on a daily basis; others are slow movers that are seldom requested. It is intuitive to leave those slow movers in the reserve area to save space for fast movers in the forward area.

In this case, it is important to carefully determine which SKUs should be assigned to the forward area and in what quantity so that the maximum benefit of the forward area can be achieved. This chapter discusses the forward reserve allocation problem, which determines the SKU assignment and space allocation in the forward area assuming it has a fixed storage capacity. Later, it will become a sub-problem in the generalized warehouse sizing and dimensioning problem discussed in Chapter 6.

Hackman and Rosenblatt (1990) proposed a mathematical model for the forward-reserve allocation problem. The model has a set of integer variables indicating whether or not a SKU is assigned to the forward area and a set of continuous variables indicating how much space is allocated for each SKU assigned to the forward area. The objective is to maximize the total benefit of the forward area, i.e., the total savings in order picking minus the total replenishing cost. The model is similar to the classical knapsack problem with the difference that it has a nonlinear objective function that is discontinuous at zero. Hackman and Rosenblatt (1990) propose a greedy heuristic to solve the forward-reserve allocation problem based on an index that ranks SKUs in terms of their desirability to be put in the forward area.

This chapter provides an alternative algorithm for the forward-reserve allocation problem that can find the guaranteed optimal solution. Extensive numerical experiments are performed to evaluate how the heuristic solutions compare with the optimal ones in terms of both the objective value and the forward assignment using problem instances based on real warehouse data. The objectives are two-fold: (1) it provides numerical justifications for using Hackman and Rosenblatt's heuristic in solving the sub-problem in the model discussed in Chapter 6; this is important since the accuracy of the sub-problem solution will greatly affect the performance of the whole algorithm; (2) it enables us to find the optimal solution of the generalized warehouse sizing and dimensioning problem.

5.2 The forward-reserve allocation problem

This section gives a brief introduction to the forward-reserve allocation model and the greedy heuristic proposed by Hackman and Rosenblatt (1990). We will illustrate the non-optimality of the heuristic through a small example and discuss its effects when the heuristic is used in a decomposition scheme to solve the sub-problem. The following notation adopted from Hackman and Rosenblatt (1990) will be used throughout this section:

Parameters:

e_i – "savings" per order picking request for SKU i if it is picked in the forward area versus in the reserve area

c_i – cost per internal replenishment

R_i – the number of requests per unit time for SKU i

D_i – the demand per unit time for SKU i converted into units of volume

N – number of SKUs in the warehouse

V – the volume of the forward area

Variables:

z_i – volume in the forward area allocated to SKU i

x_i – binary decision variable determining if SKU i is assigned to the forward area

The forward-reserve allocation model is then:

$$\text{(P1)} \qquad \max \sum_{i=1}^{N} f_i(z_i)$$

$$s.t. \quad \sum_{i=1}^{N} z_i \le V$$

$$z_i \ge 0$$

where

$$f_i(z_i) = \begin{cases} e_i R_i - \dfrac{c_i D_i}{z_i} & if \ z_i > 0 \\ 0 & if \ z_i = 0 \end{cases} \tag{5.1}$$

To simplify notation, let $a_i = e_i R_i$ and $b_i = c_i D_i$, then:

$$f_i(z_i) = \begin{cases} a_i - \dfrac{b_i}{z_i} & if \ z_i > 0 \\ 0 & if \ z_i = 0 \end{cases} \tag{5.2}$$

Problem (P1) is similar to the classical knapsack problem with the additional difficulty that $f_i(z_i)$ is nonlinear and discontinuous at zero. Hackman and Rosenblatt (1990) proposed an index (i.e., $L_i = a_i / \sqrt{b_i}$) to measure an SKU's desirability to be assigned to the forward area, and based on the index, developed the following simple heuristic to solve the problem:

Step 1: Sort the SKUs so that

$$L_i \ge L_{i+1}, \quad \forall\, i = 1, 2, \ldots, N-1$$

Step 2: For each ordered set of items $S_k = \{1, 2, \ldots, k\}$ where $1 \le k \le N$, solve problem (P1) by assuming the forward area contains only the items in S_k. Note that problem (P1) is easy to solve if the items stored in the forward area are known (see discussions in the next section).

Step 3: Select the set from all the ordered set S_k $(1 \le k \le N)$ that has the maximum objective value $v(S_k)$.

Steps 2 and 3 of the above algorithm require checking all N ordered subsets to find the one that has the maximum value of $v(S_k)$. A more efficient implementation can be developed by exploiting the fact that the function $v(S_k)$ is unimodal for $1 \le k \le N$ (see Proposition 1 in Hackman and Rosenblatt (1990)), and therefore a bisection search on k will quickly find the solution.

Bartholdi and Hackman (2005) show that the above heuristic will produce a solution that is no farther from optimum than the net-benefit of a single SKU. However, this gives no predetermined performance bound and the actual optimality gap maybe quite big as shown by the following small example. The problem has 3 SKUs to be considered for forward storage. The saving per pick is \$1 if a SKU is stored in the forward area and the cost per internal replenishment is \$40. The numbers of picks per unit time for the three SKUs (SKU1, SKU2, SKU3) are 86, 644, and 245 respectively, and the demand per unit time for the three SKUs (SKU1, SKU2, SKU3) are 122.8, 10449, and 1513.8 cubic feet respectively. The size of the forward area is 804 cubic feet. It can be verified that the heuristic will produce a solution that has an objective value of 91 with SKU1 and SKU2 in the forward area, and the optimal solution has an objective value of 207 with SKU 1 and SKU 3 in the forward area. In this particular case there is a 56% optimality gap between the heuristic and optimal solutions. It can be expected that the optimality gap will become smaller as the number of SKUs increases. Hence the heuristic algorithm will provide satisfactory solutions for practically sized problems. However, one should be more cautious when the heuristic is employed in a decomposition scheme for solving a sub-problem as the case in Chapter 6. The reason is that the master algorithm uses responses of the sub-problems to determine its search direction, and this direction is very sensitive to the response values. Even a small change (e.g., 1%) in response value can dramatically change the search direction and therefore lead to an inappropriate termination of the algorithm or significantly increases the computation time.

5.3 An optimal branch-and-bound algorithm based on outer approximation

In this section we develop an alternative algorithm to find the optimal solution for the forward-reserve allocation problem. For a given set of values for the binary variables $x \in B^N$, the forward-reserve allocation problem reduces to determining the space allocation in the forward area for those items with $x_i = 1$. If we let $X^+ = \{i : x_i = 1\}$, the sub-problem for a fixed $x \in B^N$ is:

$$v(x) = \max \sum_{i \in X^+} (a_i - \frac{b_i}{z_i})$$

$$s.t. \quad \sum_{i \in X^+} z_i \le V$$

$$z_i \ge 0, \quad \forall i \in X^+$$

Since $v(x)$ is concave in z, its optimal value can be determined from its Lagrangian dual, i.e.,

$$v(x) = \min_{u \ge 0} \max_{z \ge 0} (\sum_{i \in X^+} (a_i - \frac{b_i}{z_i}) + u(V - \sum_{i \in X^+} z_i))$$

The Lagrangian dual can be solved analytically, and we obtain:

$$v(x) = \sum_{i \in X^+} a_i - \frac{(\sum_{i \in X^+} b_i^{1/2})^2}{V} \tag{5.3}$$

This result can be written equivalently as:

$$v(x) = \sum_{i=1}^{N} x_i a_i - \frac{(\sum_{i=1}^{N} x_i b_i^{1/2})^2}{V} \tag{5.4}$$

Therefore, the original problem P1 becomes:

$$\max_{x \in B^N} v(x)$$

which is a binary nonlinear problem. We will develop a branch-and-bound algorithm based on outer approximation to solve it (see also Ryoo and Sahinidis (1996)). First, the above problem can be restated in the following equivalent form:

$$(\mathrm{P_M}) \quad \max \sum_{i=1}^{N} x_i a_i - \frac{w_1}{V}$$

$$s.t. \quad w_1 = w_2^2$$

$$w_2 = \sum_{i=1}^{N} x_i b_i^{1/2}$$

$$x_i \in [0, 1] \quad \forall i \in N$$

A linear relaxation of this problem can be developed by relaxing the nonlinear constraint $w_1 = w_2^2$ as follows. Suppose the variable w_2 has a lower and upper bound: w_2^L and w_2^U. In our case, we can take 0 and $\sum_{i=1}^{N} b_i^{1/2}$ as the respective lower and upper bound. A linear relaxation of $w_1 = w_2^2$ for $w_2 \in [w_2^L, w_2^U]$ is represented by the following set of inequalities:

$$w_1 \geq 2w_2^U w_2 - (w_2^U)^2 \tag{5.5}$$

$$w_1 \geq 2w_2^L w_2 - (w_2^L)^2 \tag{5.6}$$

$$w_1 \leq w_2^L w_2 + w_2^U w_2 - w_2^L w_2^U \tag{5.7}$$

A relaxation of P_M for $w_2 \in [w_2^L, w_2^U]$ can be represented by the following mixed integer problem P_R. Figure 5.1 illustates the linear relaxation of $w_1 = w_2^2$ for $w_2 \in [w_2^L, w_2^U]$, where the shaded area is the relaxed region bounded by the three linear constraints, i.e., (5.5, 5.6, and 5.7).

$$(P_R) \qquad \max \sum_{i=1}^{N} x_i a_i - \frac{w_1}{V} \tag{5.8.1}$$

$$s.t. \quad w_2 = \sum_{i=1}^{N} x_i b_i^{1/2} \tag{5.8.2}$$

$$w_1 \geq 2w_2^U w_2 - (w_2^U)^2 \tag{5.8.3}$$

$$w_1 \geq 2w_2^L w_2 - (w_2^L)^2 \tag{5.8.4}$$

$$w_1 \leq w_2^L w_2 + w_2^U w_2 - w_2^L w_2^U \tag{5.8.5}$$

$$w_2^L \leq w_2 \leq w_2^U \tag{5.8.6}$$

$$x_i \in [0, 1] \tag{5.8.7}$$

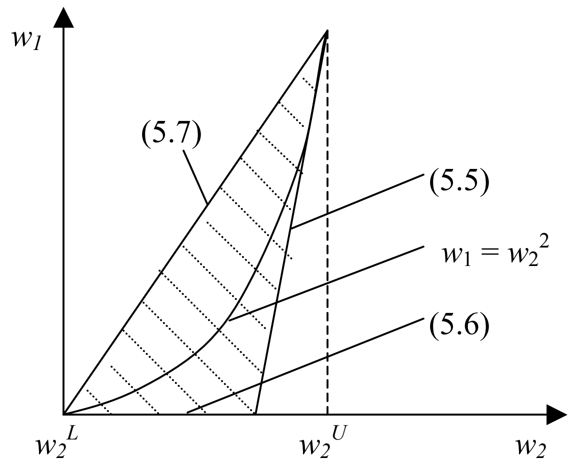


Figure 5.1 Illustration of the outer-approximation

The following proposition characterizes the optimal solution of P_R, and provides a lower and upper bound for P_M over $[w_2^L, w_2^U]$.

Proposition 1. If (x', w_1', w_2') is an optimal solution of P_R over the interval $[w_2^L, w_2^U]$, then it must satisfy $w_1' = \max(2w_2^U w_2' - (w_2^U)^2,\ 2w_2^L w_2' - (w_2^L)^2)$. An upper and lower bound of P_M over the specified interval is provided by $f(x', w_1', w_2')$ and $f(x', (w_2')^2, w_2')$ respectively, where f is the objective function of P_R (and P_M).

Proof: Let $c = \max(2w_2^U w_2' - (w_2^U)^2,\ 2w_2^L w_2' - (w_2^L)^2)$. If (x', w_1', w_2') is an optimal solution of P_R, then $w_1' \geq c$ because it must satisfy (5.8.3) and (5.8.4). Suppose $w_1' > c$, it is easy to check that (x', c, w_2') is a feasible solution of P_R, and $f(x', c, w_2') > f(x', w_1', w_2')$. Therefore, (x', w_1', w_2') cannot be optimal, which is a contradiction.

Since $(x^{'}, w_1^{'}, w_2^{'})$ is an optimal solution of P_R, $f(x^{'}, w_1^{'}, w_2^{'})$ provides an upper bound for P_M because P_R relaxes P_M. On the other hand, since $(x^{'}, w_1^{'}, w_2^{'})$ is an optimal solution of P_R, it must satisfy (5.8.2) and (5.8.7). Therefore, $(x^{'}, (w_2^{'})^2, w_2^{'})$ is a feasible solution of P_M, which provides a lower bound for P_M.

Two situations could arise for the optimal solution of P_R over any interval $[w_2^L, w_2^U]$, i.e., $w_2^{'} = w_2^L$ or w_2^U, or $w_2^L < w_2^{'} < w_2^U$. In the first case, the lower and upper bounds for P_M over $[w_2^L, w_2^U]$ are equal since the relaxation is tight at the end points of the interval. In the second case, the lower and upper bounds for P_M over $[w_2^L, w_2^U]$ are not equal, and therefore the previous relaxation needs to be further refined to provide a more precise approximation for P_M. From Proposition 1, $(w_1^{'}, w_2^{'})$ always lies on the boundary defined by the two linear functions (i.e., (5.8.3) and (5.8.4)) that underestimate the function $w_1 = w_2^2$. A better approximation can be constructed by dividing $[w_2^L, w_2^U]$ into two sub-intervals $[w_2^L, w_2']$ and $[w_2', w_2^U]$, and developing outer approximations on each of the sub-intervals (as illustrated by the shaded area in Figure 5.2, note that the previous solution $(x^{'}, w_1^{'}, w_2^{'})$ is already cut off). Based on this idea, a branch-and-bound procedure can be developed to solve P_M optimally by recursively dividing the original interval of w_2 into smaller sub-intervals to provide more accurate approximations of P_M. At any iteration of the branch-and-bound procedure, a list of sub-intervals is maintained that define the current approximation of P_M. The algorithm terminates if the optimality gap is sufficiently small; otherwise, one of the sub-intervals is selected and further divided to provide a refined approximation of $w_1 = w_2^2$.

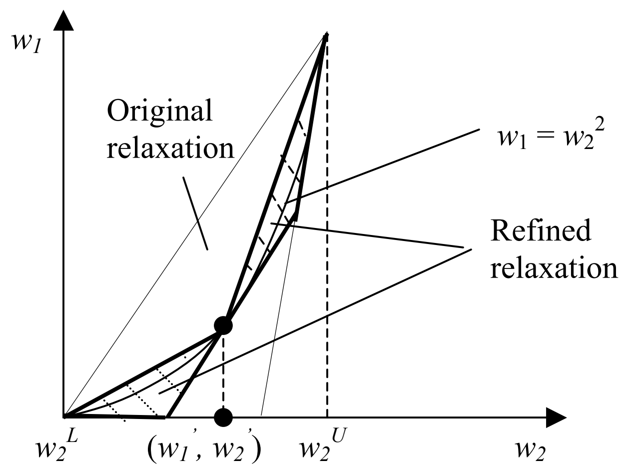


Figure 5.2 Illustration of the branch-and-bound procedure

Let (x^*, w_1^*, w_2^*) be the optimal solution of P_M. Denote $\bigcup_{i \in I} [(w_2^L)_i, (w_2^U)_i]$ as the current set of sub-intervals in the w_2 space that contains w_2^* and defines the current approximation of P_M. Let UB_i and LB_i be the local upper and lower bounds for P_M over subinterval i, and UB and LB be the global upper and lower bounds. We have the following relations:

$$\max\{LB_i \mid i \in I\} \le f(x^*, w_1^*, w_2^*) \le UB_{i|(w_2^L)_i \le w_2^* \le (w_2^U)_i} \le \max\{UB_i \mid i \in I\}$$

The first inequality in the above equation is due to the fact that each LB_i corresponds to a feasible solution of P_M and therefore is less than or equal to the optimal solution. The second inequality is because if $w_2^* \in [(w_2^L)_i, (w_2^U)_i]$ for any $i \in I$ (in other words there exits an i for which this is true), then the optimal solution is less than or equal to the relaxed optimal solution of P_M over that sub-interval, i.e., UB_i. Therefore, the global upper and lower bounds are given by:

$$LB = \max\{LB_i \mid i \in I\} \text{ and } UB = \max\{UB_i \mid i \in I\}$$

The branch-and-bound algorithm for solving P_M is formally stated as follows. Define set *I* as a list of the sub-intervals in the w_2 space. Each sub-interval has a relaxed optimal solution and the lower and upper bounds (i.e., UB_i and LB_i) for P_M over that sub-interval.

Branch-and-Bound Algorithm

1. Initialization: select the convergence tolerance parameter $\varepsilon > 0$; define the initial bound $[w_2^L, w_2^U]$ for w_2; solve P_R over $[w_2^L, w_2^U]$ to obtain the initial global lower and upper bound: LB and UB; define the set *I*, which initially contains only $[w_2^L, w_2^U]$ with its associated relaxed optimal solution and bounds.
2. Termination test: if $UB - LB \le \varepsilon$, then terminate and the solution that yields the current global lower bound (i.e., the best feasible solution) is optimal.
3. Branch and Bound: remove from *I* the interval $[(w_2^L)_i, (w_2^U)_i]$ that has the maximum upper bound (i.e., the interval that defines the current global upper bound); divide $[(w_2^L)_i, (w_2^U)_i]$ into two sub-intervals $[(w_2^L)_i, (w_2^{'})_i]$ and $[(w_2^{'})_i, (w_2^U)_i]$, solve P_R over the sub-intervals to obtain the relaxed optimal solution as well as the lower and upper bound for P_M over the corresponding sub-intervals, and insert the sub-intervals into *I*; Update $LB = \max\{LB_i : i \in I\}$ and $UB = \max\{UB_i : i \in I\}$; delete all intervals that satisfy $UB_i < LB$ from *I*; go to Step 2.

In the above algorithm, if the termination criterion is not satisfied, we select an interval and further divide it into two smaller intervals in hope of finding better bounds. This explains why the interval that has the maximum upper bound among all intervals currently in I is selected as the candidate for branching; it defines the current global upper bound ($UB = \max\{UB_i : i \in I\}$) and by branching on it we hope to reduce the global upper bound. The following proposition shows that the branch-and-bound algorithm will converge to the optimal solution in a finite number of iterations.

Proposition 2: The branch-and-bound algorithm will converge to the optimal solution after a finite number of branchings on w_2.

Proof: In step 3 of the branch-and-bound algorithm, an interval $[w_2^L, w_2^U]$ is selected and branched into two sub-intervals $[w_2^L, w_2^{'}]$ and $[w_2^{'}, w_2^U]$, where $w_2^{'}$ corresponds to the optimal solution of P_R for $w_2 \in [w_2^L, w_2^U]$. Therefore, it satisfies $w_2^{'} = \sum_{i=1}^{N} x_i b_i^{1/2}$ for a certain $x \in B^N$ due to constraint (5.8.2). Since x is a discrete variable, there are only a finite number of possible values for $w_2^{'}$, or in other words, the interval can only be branched into a finite number of sub-intervals according to the algorithm.

If a sub-interval $[(w_2^L)_i, (w_2^U)_i]$ cannot be further branched, it means we cannot find a relaxed optimal solution that satisfies $(w_2^L)_i < (w_2^{'})_i < (w_2^U)_i$. Therefore, the relaxed optimal solution over $[(w_2^L)_i, (w_2^U)_i]$ must satisfy $(w_2^{'})_i = (w_2^L)_i$ or $(w_2^U)_i$. Because the relaxation is tight at the end points of the intervals, the lower and upper bounds on the interval $[(w_2^L)_i, (w_2^U)_i]$ are equal (i.e., $UB_i = LB_i$).

So after a finite number of iterations, the algorithm will terminate either because the optimality gap is sufficiently small or all sub-intervals cannot be further branched. In the latter case, the global upper ($UB = \max\{UB_i : i \in I\}$) and lower ($LB = \max\{LB_i : i \in I\}$)

bounds must be equal since $UB_i = LB_i$ for all i. Therefore, the branch-and-bound algorithm converges to the optimal solution after a finite number of branching.

5.4 Computational results

This section provides numerical results that demonstrate the computational performance of the proposed algorithm and compare the heuristic and optimal solutions using a set of practical examples.

5.4.1 Test problems

Test problems used in the numerical experiments are generated based on two basic data sets from real warehouses as provided by Bartholdi and Hackman (2005). The first data set (S1) is from an office product warehouse and the second (S2) is from a tire warehouse. Table 5.1 shows the summary statistics of these two data sets. It can be seen that these two data sets represent quite different warehouse scenarios as seen from the statistics of L_i, i.e, the ranking index measuring a SKU's desirability to be assigned to the forward area used by the heuristics algorithm. This difference is mainly due to the fact that the average picking size is much smaller in the office product warehouse (e.g., staplers and clips) than that in the tire warehouse (e.g., tires). For each scenario, samples are randomly generated with different sizes (i.e., N = 50, 100, 500, 1000, 5000, and 10000 SKUs) following the same distribution (frequency histogram) of a_i and b_i in the basic data set. The size of the forward area is set at three difference levels (i.e., V_1, V_2, and V_3) for each scenario and each sample size so that there are approximately 20%, 50%, and 80% SKUs assigned to the forward area in the optimal solution. In summary, there are totally 36 cases (2×6×3) with different warehouse scenarios, different numbers of SKUs, and/or different sizes of the forward area. For each case, 50 instances are randomly generated that gives a total of 1800 testing problems.

Table 5.1 Summary statistics for the two basic data sets

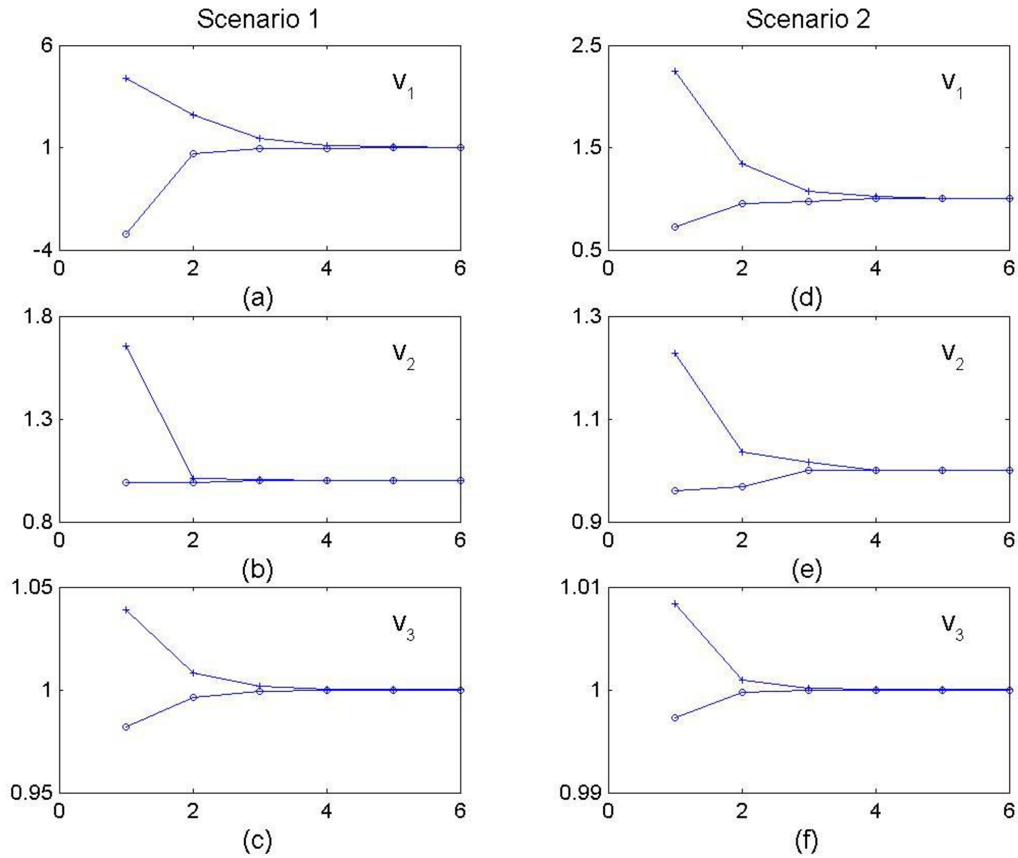
		Mean	Median	StDev	Minimum	Maximum
S1	a_i	18.85	14.5	12.88	1.5	55.5
	b_i	17.45	10.56	18.43	0.28	90.1
	L_i	5.219	4.74	2.517	0.873	11.3
S2	a_i	27.521	15.309	30.49	0.945	186
	b_i	927.7	470.4	1238.7	11.2	7661
	L_i	0.88145	0.76421	0.41755	0.08929	2.28

5.4.2 Computational efficiency of the optimal algorithm

The proposed algorithm is implemented in C, which calls ILOG\CPLEX to solve the relaxed problem P_R. All tests are performed on a Sun 280R server with 2×900MHz UltraSparc-III CPU and 2GB RAM. Table 5.2 shows the average and range of computation time for the different testing cases (each has 50 randomly generated problem instances). In general, the algorithm is very efficient and in most cases can converge to the optimal solution within 60 seconds. The results in Table 5.2 also suggest that the computation time is much shorter for cases with a larger forward area. A detailed look at the convergence history of the algorithm shows that for otherwise identical parameters, increasing the size of the forward area usually results in a smaller initial optimality gap, as illustrated in Figure 5.3. Figures 5.3(a, b, c) (i.e., the three figures on the left) show the convergence history of the optimal algorithm for scenario 1 (S1) with 5000 SKUs, and Figures 5.3(d, e, f) (i.e., the three figures on the right) for scenario 2 (S2) with 5000 SKUs. The two lines in each sub-figure represent the normalized upper and lower bounds (i.e., the actual bounds divided by the corresponding optimal value). It can be seen that the algorithm gives a very tight bound after the first iteration for both scenarios when the size of the forward area is set at *V3*. Figure 5.3 also suggests that the algorithm can quickly locate a near optimal solution within a few iterations. For example, the relative optimality gap is within 0.01% of the optimal value after 5 iterations for all cases shown in Figure 5.3. Similar results were found in all other tested cases.

Table 5.2 Computational time of the optimal algorithm (seconds)

		50 SKUs	100 SKUs	500 SKUs	1K SKUs	5K SKUs	10K SKUs
S1	V_1	0.14 (0.05, 0.25)	0.34 (0.13, 0.56)	4.40 (1.99, 7.76)	8.33 (4.21, 16.54)	64.00 (25.1, 153.9)	135.68 (15.9, 314.1)
	V_2	0.07 (0.03, 0.15)	0.12 (0.05, 0.23)	1.08 (0.56, 2)	2.03 (1.17, 3.23)	6.63 (1.88, 15.1)	10.3 (4.04, 37.76)
	V_3	0.02 (0.01, 0.05)	0.04 (0.02, 0.07)	0.41 (0.26, 0.53)	0.95 (0.7, 1.26)	1.53 (0.99, 2.13)	3.49 (2.73, 4.79)
S2	V_1	0.09 (0.05, 0.16)	0.18 (0.1, 0.29)	2.04 (1.13, 3.35)	3.83 (2.16, 6.96)	18.50 (6.57, 56.79)	31.01 (6.12, 91.14)
	V_2	0.05 (0.01, 0.11)	0.10 (0.05, 0.17)	1.07 (0.56, 1.67)	1.65 (0.91, 2.64)	3.84 (1.43, 7.47)	4.71 (3.06, 9.99)
	V_3	0.02 (0.01, 0.04)	0.03 (0.01, 0.06)	0.30 (0.21, 0.46)	0.71 (0.56, 0.99)	0.98 (0.72, 1.27)	2.32 (1.79, 3.15)



Note: The vertical axes are scaled differently in order to clearly show the gaps in different cases.

Figure 5.3 Convergence of the optimal algorithm with 5000 SKUs

5.4.3 Comparing the optimal and heuristic solutions

The optimal algorithm not only provides an alternative method to solve the forward-reserve problem, but also allows us to evaluate the optimality of the heuristics by comparing the heuristic and optimal solutions for practical problems. Table 5.3 shows the number of times that the heuristic objective value coincides with the optimum within a calculation precision of $\pm 10^{-3}$ for the 50 randomly generated instances of each tested case. The results suggest that the heuristic solution can often find the optimal solution (within a precision of $\pm 10^{-3}$). For instances where an optimal solution is not found by the heuristic, the actual optimality gap is always very small. Table 5.4 shows the maximum relative optimality gap (i.e., the absolute gap divided by the corresponding optimal value) for all instances that the heuristics failed to find the optimal solution. It can be seen that even for cases with 50 SKUs and a forward area size of *V1* (i.e., approximately 10 SKUs are assigned to the forward area), the relative optimality gap is very small: less than 0.313% for the office product warehouse and less than 0.039% for the tire warehouse. The relative gap becomes even smaller as the number of SKUs increases. Besides comparing the objective value, we also compared the optimal and heuristic solution in terms of their forward assignment (i.e., x_i). In order to do this, we use the difference index DI to measure the similarity of two solutions, which is defined as the ratio of the number of SKUs that have different assignment in the optimal and heuristic solutions (i.e., x_i not equal in the optimal and heuristic solutions) and the total number of SKUs. The smaller the index value is, the more similar the two solutions are. Table 5.5 shows the maximum DI over the 50 randomly generated instances of each test case. Note that it is possible that two solutions have the same objective value but different forward assignments. The results suggest that the heuristic solution is very close to the optimal in terms of the forward assignment. Even for cases with 10000 SKUs, there are less than 5 SKUs (10000 $\times$ 0.0005) that are different in terms of the optimal and heuristic assignments. In summary, although the heuristic may produce a large gap in some small examples, the solutions when it is applied to larger practical problems are always very close to the optimal in terms of both the objective value and the forward assignment. This demonstrates that the ranking index L_i is a very effective measure in selecting the set of SKUs to assign to the forward area.

Table 5.3 Number of times (out of 50) that the heuristic solution is optimal

		50 SKUs	100 SKUs	500 SKUs	1K SKUs	5K SKUs	10K SKUs
S1	V_1	44	40	44	40	45	42
	V_2	47	49	50	42	41	43
	V_3	50	50	46	47	38	50
S2	V_1	48	47	46	47	49	43
	V_2	48	50	50	44	47	43
	V_3	49	50	50	44	37	39

Table 5.4 Maximum relative optimality gap (%)

		50 SKUs	100 SKUs	500 SKUs	1K SKUs	5K SKUs	10K SKUs
S1	V_1	3.13E-01	2.37E-01	5.50E-03	1.10E-03	5.70E-06	3.32E-05
	V_2	4.91E-02	1.20E-02	0	9.19E-06	1.84E-06	1.78E-06
	V_3	0	0	5.13E-05	4.40E-06	1.11E-06	0
S2	V_1	3.90E-02	5.86E-03	5.39E-04	5.07E-06	7.82E-07	2.81E-04
	V_2	7.95E-03	0	0	3.52E-06	5.97E-07	2.53E-05
	V_3	3.81E-04	0	0	3.78E-06	7.13E-07	3.68E-06

Table 5.5 Maximum DI of the optimal and heuristic solutions

		50 SKUs	100 SKUs	500 SKUs	1K SKUs	5K SKUs	10K SKUs
S1	V_1	0.04	0.02	0.004	0.002	0.0002	0.0001
	V_2	0.02	0.01	0	0.005	0.0004	0.0002
	V_3	0	0.02	0.004	0.001	0.0002	0
S2	V_1	0.04	0.01	0.004	0.002	0.0006	0.0005
	V_2	0.02	0	0.002	0.001	0.001	0.0004
	V_3	0.04	0	0	0.002	0.0004	0.0004

5.5 Conclusions

This chapter develops a branch-and-bound algorithm based on outer approximation to optimally solve the forward-reserve allocation problem. The outer approximation method is different from the piecewise linearization method (SOS2) in that, at each iteration, it can provide a relaxation of the original nonlinear model and therefore a upper bound to the optimal solution. This, combined with the lower bound obtained from a feasible solution, enables us to use the branch and bound scheme to iteratively find a guaranteed optimal solution. Computational results demonstrate that the proposed algorithm is effective in solving the problem such that the optimal solution can be found in less than 60 seconds for most of the realistically sized problem instances. The heuristic solutions based on raking of the SKUs are compared with the optimal solutions in terms of both the objective value and the forward assignment using problem instances based on real warehouse data. The results suggest that the greedy heuristic solutions for practical problems are so close to the optimum that the difference can almost be ignored as rounding errors.

CHAPTER 6

THE SIZING AND DIMENSIONING PROBLEM WITH FORWARD RESERVE ALLOCATION

6.1 Introduction

This chapter presents a decision model for the sizing and dimensioning problem incorporating the decision of forward reserve allocation. The decisions in the model include: (1) the warehouse size and dimension; (2) the space allocation between the forward and reserve area; (3) the SKU assignment to the forward area and in what quantity; and (4) the space allocation in the reserve area. The objective is to minimize the total cost of equipment, inventory, and material handling for order picking, internal replenishment, and put-away.

A simplified version of this problem was discussed in Chapter 4, where the forward area is assumed to hold all SKUs so that order picking is performed only in the forward area. This restriction enables the problem to be solved optimally with a Generalized Benders Decomposition method. The general problem has the additional decision of whether to assign an SKU to the forward area or not. This warehouse assignment problem (without sizing decision) is shown by Hackman and Rosenblatt (1990) to be NP-complete. In this chapter we propose a two-level hierarchical heuristic algorithm to solve the general problem in which the assignment of SKUs to the forward area together with the warehouse sizing and dimensioning are decision variables.

The remaining sections of this chapter are organized as follows: Section 6.2 develops an integrated mathematical model for the generalized forward reserve sizing and dimensioning problem; Section 6.3 presents an efficient hierarchical heuristic algorithm to solve the problem; Section 6.4 gives numerical results with regards to the performance of the proposed heuristic algorithm; and Section 6.5 summarizes the results and conclusions.

6.2 Mathematical models

We follow the discussion in section 4.2, but relax the assumption that all SKUs are included in the forward area. It is assumed that if an SKU is assigned to the forward area, all customer requests for that SKU are fulfilled from the forward area. Otherwise, the SKU is assigned only to the reserve area and customer requests are fulfilled from the reserve area. Many of the notations are adopted from section 4.2, and therefore will not be repeated here. The following additional notations will be used in the generalized model.

Parameters:

C_{o1} – order picking cost per unit of travel distance for order picking in the forward area

C_{o2} – order picking cost per unit of travel distance for order picking in the reserve area

N_f – average number of picks per order picking tour in the forward area

N_b – average number of picks per order picking tour in the reserve area

R_i – annual number of picks for SKU i

W – width of the cross aisle in the middle of the forward area (see Figure 6.1)

Variables:

x_i – binary decision variable determining if SKU i is assigned to the forward area

The total life-cycle cost of the warehouse is the sum of the construction cost for space and equipment and the net present value of the discounted operational cost for order picking, internal replenishment, put-away, inventory holding, and external replenishment. The cost models are discussed in the following paragraphs.

(1) Construction cost

The warehouse layout shown in Figure 6.1 is slightly different from that in Figure 4.1 in that there is a wide cross aisle in the middle of the forward area. It is used to allow material flow from the reserve area to the shipping dock for orders picked in the reserve area.

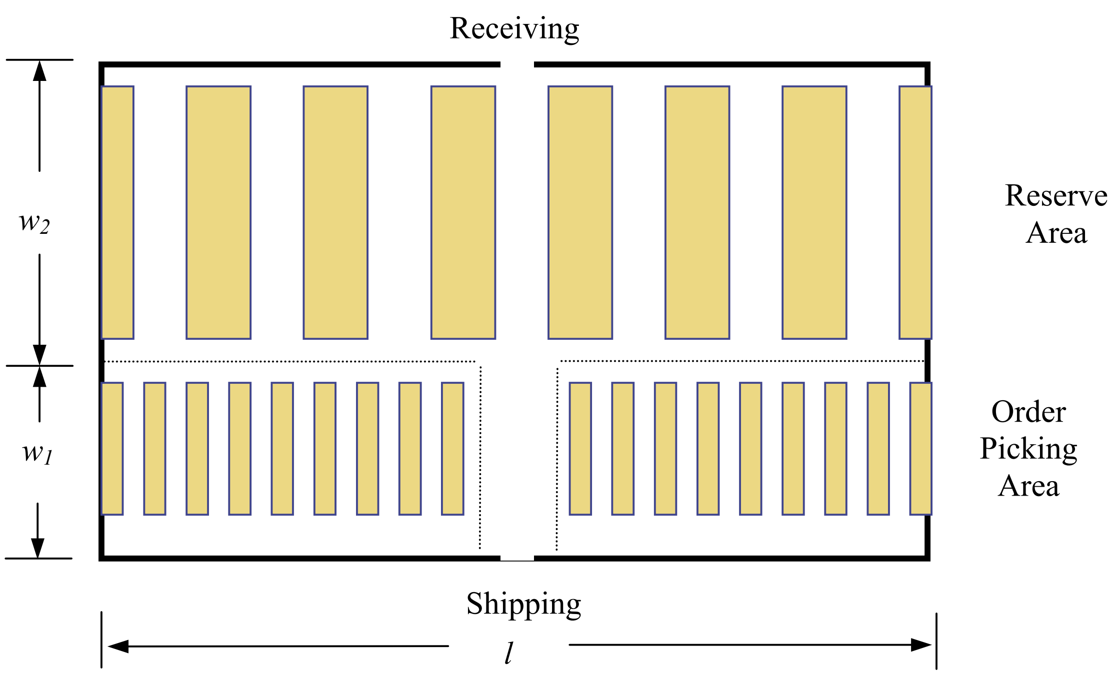


Figure 6.1 A block layout of the forward-reserve warehouse

Warehouse construction cost is modeled as a function of the warehouse area and perimeter. The warehouse dimensions are given by:

$$l = y_l A_f + W; \quad w_1 = y_{w1} A_s; \quad w_2 = y_{w2} A_p$$

Warehouse construction cost is then modeled following White and Francis (1971) and Bassan et al. (1980) as:

$$C_{Const} = C_{a1} l w_1 + C_{a2} l w_2 + 2C_w (l + w_1 + w_2) \tag{6.1}$$

(2) Material handling cost

The material handling cost in the warehouse includes the cost of put-away, internal replenishment, and order picking.

<u>Cost of put-away</u>

The cost of put-away is the same as discussed in Section 4.2, which is

$$C_{PutAway} = C_p (\frac{l}{2} + w_2)(\sum_i \frac{D_i}{V_p}) \tag{6.2}$$

where the first two factors represent the average cost per put-away trip and the last factor represents the total number of put-away trips per year.

<u>Cost of internal replenishment</u>

The cost of internal replenishment is derived following the same method as discussed in Section 4.2. However, the representation for the average interval replenishment cost per trip (Equation (6.3)) is different to account for the cross aisles in the middle of the forward area.

$$c_r = C_r (\frac{16l^3 - W^3 - 3l^2 W}{24l(l - W)} + w_1 + w_2) \tag{6.3}$$

If we let $W = 0$, it can be seen that the above formula is equivalent to Equation (4.3). The total annual internal replenishment cost can then be calculated as:

$$C_{InRpl} = c_r(\sum_i \frac{x_i D_i}{z_i}) \tag{6.4}$$

Note that the last term represents the annual number of internal replenishments for all SKUs assigned to the forward area as indicated by the assignment variables x_i.

<u>Cost of order picking</u>

Orders are batch picked using the traversal routing policy from the forward area for SKUs assigned to the forward area, and from the reserve area for SKUs not assigned to the forward area. If an order consists of SKUs to be picked from both areas, the picked items need to be consolidated before the order can be shipped to the customer. The consolidation cost is not captured in this model since it is independent of the sizes and dimensions of the forward and reserve areas. The average cost per batch picking in the forward area is modeled similarly to Equation (4.4) with the additional factor $2W$ to account for the effects of the cross aisle in the middle of the forward area:

$$C_{o1}\left(2l\left(\frac{N_f - 1}{N_f + 1}\right) + w_1 y_l\left(1 - \left(1 - \frac{1}{y_l}\right)^{N_f}\right) + 0.5w_1 + 2W\right)$$

The annual cost for order picking in the forward area is then:

$$C_{PickF} = \frac{\sum_i x_i R_i}{N_f} \times C_{o1}\left(2l\left(\frac{N_f - 1}{N_f + 1}\right) + w_1 y_l\left(1 - \left(1 - \frac{1}{y_l}\right)^{N_f}\right) + 0.5w_1 + 2W\right) \tag{6.5}$$

The average cost per batch picking in the reserve area is estimated by:

$$C_{o2}\left(2l\left(\frac{N_b - 1}{N_b + 1}\right) + \frac{lw_2}{A_b}\left(1 - \left(1 - \frac{A_b}{l}\right)^{N_b}\right) + 0.5w_2 + 2w_1\right)$$

The factor $2w_1$ represents the distance for crossing the forward area in order to pick SKUs in the reserve area. The annual cost for order picking in the reserve area is given by:

$$C_{PickR} = \frac{\sum_i (1-x_i)R_i}{N_b} \times C_{o2}\left(2l\left(\frac{N_b-1}{N_b+1}\right) + \frac{lw_2}{A_b}\left(1-\left(1-\frac{A_b}{l}\right)^{N_b}\right) + 0.5w_2 + 2w_1\right) \qquad (6.6)$$

The above models are based on the layout shown in Figure 6.1 with randomized storage and traversal routing policy. Other layout structures can be used as well, for example, the aisles can be oriented horizontally as shown in Figure 4.3. Other storage and routing policies can be employed as well, for example, dedicated storage and/or return routing policy. Travel cost models for these variations are discussed in Caron et al. (1998).

(3) Inventory holding and external replenishment cost

An external replenishment for a SKU is performed when its total inventory in the warehouse drops to a constant safety stock level S_i that is given based on the replenishment lead time and product demand. The total annual inventory holding and external replenishment cost is represented by (6.7) following the standard EOQ model:

$$C_{Invt} = \sum_i (A_i \frac{D_i}{q_i} + C_i(\frac{q_i}{2} + S_i)) \qquad (6.7)$$

(4) The mathematical model

The problem can be described as follows: Determine the dimensions of the warehouse, the assignment of SKUs to the forward area, and the space allocation in both the forward and reserve areas, to minimize the total warehouse life-cycle cost including the construction cost and the net present value of the discounted operational cost, subject to the storage capacity constraints in both the forward and reserve areas. The optimization model can be stated as:

$$\text{(P)} \qquad min\ C_{Construct} + (C_{PutAway} + C_{InRpl} + C_{PickF} + C_{PickR} + C_{Invt}) \times Z \tag{6.8.1}$$

$$\text{s.t.} \quad \sum_i (q_i + S_i) \le \beta_b l w_2 \tag{6.8.2}$$

$$\sum_i x_i z_i \le \beta_f l w_1 \tag{6.8.3}$$

$$q_i, z_i \ge 0, \forall\, i \in [1, N] \tag{6.8.4}$$

$$x_i \in \{0, 1\}, \forall\, i \in [1, N] \tag{6.8.5}$$

$$y_l, y_{w1}, y_{w2} \in Z_+^3 \tag{6.8.6}$$

Constraints (6.8.2) and (6.8.3) are the storage space constraints in the reserve and forward area respectively. Section 4.2 gave a detailed discussion of these constraints. The quadratic term in (6.8.3) ensures that the storage space requirement is calculated only for SKUs that are assigned to the forward area (i.e., $x_i = 1$). Additional constraints can be added to the above model, for example, constraints on the dimension variables due to construction site limits and/or warehouse shape ratio limits, and constraints on the space allocation variables representing lower and upper bounds for the space allocation of a single SKU and/or a group of SKUs. The same algorithm can be applied with slight modifications as discussed in the next section. The model incorporates the binary assignment variables x_i to indicate whether or not a product is assigned to the forward area. The number of binary and continuous variables is each roughly proportional to the number of SKUs. For realistic problem instances this implies that the number of variables will equal the tens of thousands.

6.3 Solution method

Problem (P) can be transformed to the following equivalent form:

$$\text{(P')} \quad min \ f_1(\bar{y}) + f_2(\bar{y}) - f_3(\bar{y}) \tag{6.9.1}$$

$$\text{s.t.} \quad \bar{y} = \{y_l, y_{w1}, y_{w2}\} \in Z_+^3 \tag{6.9.2}$$

where:

$$\begin{aligned} f_1(\bar{y}) = {} & C_{a1}(A_f y_l + W)A_s y_{w_1} + C_{a2}(A_f y_l + W)A_p y_{w_2} \\ & + 2C_w(A_f y_l + W + A_s y_{w_1} + A_p y_{w_2}) \\ & + C_p(\sum_i \frac{D_i}{V_p})Z(\frac{A_f y_l + W}{2} + A_p y_{w_2}) \\ & + \frac{\sum_i R_i}{N_b} C_{o2} Z \times \left(\begin{array}{l} 2(A_f y_l + W)\left(\frac{N_b - 1}{N_b + 1}\right) + \frac{(A_f y_l + W)A_p y_{w_2}}{A_b} \\ \left(1 - \left(1 - \frac{A_b}{A_f y_l + W}\right)^{N_b}\right) + 0.5A_p y_{w_2} + 2A_s y_{w_1} \end{array} \right) \end{aligned} \tag{6.10}$$

and $f_2(\bar{y})$, $f_3(\bar{y})$ are the solutions to the following sub-problems:

$$\text{(P1)} \quad f_2(\bar{y}) = \min_{q_i} \sum_i Z(A_i \frac{D_i}{q_i} + C_i \frac{q_i}{2}) \tag{6.11.1}$$

$$\text{s.t.} \quad \sum_i q_i \le \beta_b(A_f y_l + W)A_p y_{w_2} - \sum_{i \in I} S_i \tag{6.11.2}$$

$$q_i \ge 0, \forall i \in [1, N] \tag{6.11.3}$$

$$\text{(P2)} \quad f_3(\bar{y}) = \max_{x_i, z_i} \sum_i x_i(eR_i - c\frac{D_i}{z_i}) \tag{6.12.1}$$

$$\text{s.t.} \quad \sum_i x_i z_i \le \beta_f(A_f y_l + W)A_s y_{w_1} \tag{6.12.2}$$

$$z_i \ge 0, \forall i \in [1, N] \tag{6.12.3}$$

$$x_i \in \{0, 1\}, \forall i \in [1, N] \tag{6.12.4}$$

Note that P2 is similar to the forward reserve allocation model discussed in Chapter 5, with the difference that the savings per request (i.e., e) if an SKU is stored in the forward area and the cost per internal replenishment (i.e., c) are no longer constants, but are functions of the dimension variables represented as follows:

$$e = \frac{C_{o2}Z}{N_b}\left(\frac{2(A_f y_l + W)\left(\frac{N_b - 1}{N_b + 1}\right) + \frac{(A_f y_l + W)A_p y_{w_2}}{A_b}}{\left(1 - \left(1 - \frac{A_b}{A_f y_l + W}\right)^{N_b}\right) + 0.5 A_p y_{w_2} + 2A_s y_{w_1}}\right) - \frac{C_{o1}Z}{N_f}\left(2(A_f y_l + W)\left(\frac{N_f - 1}{N_f + 1}\right) + A_s y_l y_{w_1}\left(1 - \left(1 - \frac{1}{y_l}\right)^{N_f}\right) + 0.5 A_s y_{w_1} + 2W\right) \tag{6.13}$$

$$c = C_r Z\left(\frac{16(A_f y_l + W)^3 - W^3 - 3(A_f y_l + W)^2 W}{24(A_f y_l + W)A_f y_l} + A_s y_{w_1} + A_p y_{w_2}\right) \tag{6.14}$$

The above transformation decomposes the original large mixed-integer nonlinear problem into a master problem and two sub-problems. The master problem (P') is a small problem consisting of only three integer variables. Sub-problem (P1) is the constrained EOQ model, which is convex and can be solved very efficiently as shown in Section 4.3.2. Sub-problem (P2) is the forward-reserve allocation problem discussed by Hackman and Rosenblatt (1990) for any fixed $\bar{y}$. It determines for a given size of the forward area the set of SKUs to be assigned to the forward area and in what quantity so that the total benefit of such an assignment is maximized. Chapter 5 shows that the heuristic proposed by Hackman and Rosenblatt (1990) produce solutions that are very close to the optimal in practical cases. Therefore, we use the heuristic presented in Section 5.2 to solve problem (P2).

Solving the master problem

Problem (P') is the master problem, which determines an optimal $\bar{y}$ to minimize the total life-cycle cost. Its objective function cannot be analytically represented due to the fact that $f_2(\bar{y})$ and $f_3(\bar{y})$ are solutions of two other optimization problems, i.e., (P1) and (P2). This section presents a pattern search based heuristic to solve the master problem, which does not require the estimation of the gradient but rather uses only the function values.

Several pattern search methods have been proposed in the past for deterministic function optimization (Torczon (1997)). The method used in this section is the Nelder-Mead simplex method, which is one of the most popular pattern search methods (Nelder and Mead (1965)). The original Nelder-Mead method is proposed for a continuous optimization problem, so problem (P') is first treated as a continuous problem, and the result will be rounded to its nearest integer solutions when the algorithm terminates. The algorithm is stated as follows (see Barton and Ivey (1996) for more details as well as some improvements to the original algorithm):

Step 1: Initialization. Choose 4 affinely independent points to form an initial 3-dimensional simplex. Evaluate the objective function $F(\bar{y}_i)$ (by solving the sub-problem) at each point $\bar{y}_i$ for i = 1, 2, …, 4.

Step 2: Stopping criterion. Iterations continue until the standard deviation of the 4 function values at the extreme points of the simplex falls below a particular value, or the size of the simplex becomes sufficiently small, or the maximum number of function evaluation is reached.

Step 3: Reflect worst point. At the start of each iteration, identify the vertices where the highest, second highest, and lowest function values occur. Let $\bar{y}_{high}$, $\bar{y}_{\sec h}$, and $\bar{y}_{low}$ denote these points respectively, and let F_{high}, $F_{\sec h}$, and F_{low} represent the corresponding function values. Find $\bar{y}_{cent}$, the centroid of all vertices other than $\bar{y}_{high}$. Generate a new vertex $\bar{y}_{refl}$ by reflecting $\bar{y}_{high}$ through $\bar{y}_{cent}$. Reflection is computed according to the following equation, where α is the reflecting coefficient ($\alpha > 0$):

$$\bar{y}_{refl} = (1+\alpha)\bar{y}_{cent} - \alpha\bar{y}_{high}$$

Step 4a: Accept reflection. If $F_{low} \le F_{refl} \le F_{\sec h}$, then $\bar{y}_{refl}$ replaces $\bar{y}_{high}$ in the simplex, and go to step 2.

Step 4b: Attempt expansion. If $F_{refl} < F_{low}$, then the reflection is expanded in the hope that more improvement will result by expanding the search in the same direction. The expansion point is calculated using the following equation, where the expansion coefficient is γ ($\gamma > 1$).

$$\bar{y}_{\exp} = (1-\gamma)\bar{y}_{cent} + \gamma\bar{y}_{refl}$$

If $F_{\exp} < F_{refl}$, then $\bar{y}_{\exp}$ replaces $\bar{y}_{high}$ in the simplex; otherwise, the expansion is rejected and $\bar{y}_{refl}$ replaces $\bar{y}_{high}$. Go to step 2.

Step 4c: Attempt contraction. If $F_{refl} > F_{\sec h}$, then the simplex contracts. If $F_{refl} \le F_{high}$, then $\bar{y}_{refl}$ replaces $\bar{y}_{high}$ and F_{refl} replaces F_{high} before attempting contraction or shrinking. The contraction vertex is calculated by the following equation, where the contraction coefficient is β ($0 < \beta < 1$).

$$\bar{y}_{cont} = (1-\beta)\bar{y}_{cent} + \beta\bar{y}_{high}$$

If $F_{cont} < F_{high}$, then the contraction is accepted and go to step 2.

Step 4c': Shrink. If $F_{cont} > F_{high}$, then the contraction has failed, and the entire simplex shrinks by a factor of δ ($0 < \delta < 1$), retaining only $\bar{y}_{low}$. This is done by replacing each extreme point $\bar{y}_i$ (except $\bar{y}_{low}$) by:

$$\bar{y}_i = (1-\delta)\bar{y}_{low} + \delta\bar{y}_i$$

The algorithm then evaluates the function value at each vertex (except $\bar{y}_{low}$) and goes to step 2.

Step 5: Termination. Round up the resulted solution to its nearest integer solution.

The above algorithm is for unconstrained optimization. It can be extended to solve problems that have bound and linear constraints (for details see Lewis and Torczon (1999, 2000)). These constraints may represent layout feasibility constraints such as construction site limits and warehouse shape ratio limits.

6.4 Numerical results

Three basic problems are tested that each has a different number of SKUs (i.e, 8000 SKUs, 15000 SKUs, and 30000 SKUs respectively). The optimal solutions of these problems are obtained using a naïve enumeration method that enumerates all the possible combinations of the dimension variables and finds the one that has the minimum objective value. In order to ensure the optimality of the enumeration method, sub-problem (P2) is solved using the optimal algorithm proposed in Chapter 5 instead of the heuristic algorithm. It should be noted that the enumeration method is very inefficient (usually takes hours and in some cases days), but it can provide the optimal solution. Two scenarios are considered: (1) all SKUs are included in the forward area; (2) a subset of the SKUs is assigned to the forward area. Table 6.1 shows the total cost, the warehouse size (i.e., $l*(w_1+w_2)$), shape ratio (i.e., $(w_1+w_2)/l$), and the ratio of the forward area and the total warehouse area (i.e., $w_1/(w_1+w_2)$) in the optimal solutions of the three tested problems under both scenarios. It can be seen that by allowing the additional flexibility of assigning SKUs to the forward area based on their flow activities, it is possible to reduce the total life-cycle cost with a smaller warehouse and a smaller forward area. However, this cost savings needs to be balanced against the additional complexity introduced by picking orders from two different areas, e.g., splitting and assembling an order picked from different areas. In general, scenario 2 is more attractive in the following situations: (1) the construction cost of the forward area is high; (2) the internal replenishing cost is high; (3) the picking activities for the SKUs are highly skewed. This is illustrated in Table 6.2, in which the construction cost per unit of the forward area (C_{a1}) and the internal replenishment cost per unit of travel distance (C_r) are increased by a factor of 1, 5 and 10 from their base level (i.e., C_{a1} = \$30/ft^2 and C_r = \$0.0028/ft), and the number of picks for SKUs (R_i) are skewed at three different level, i.e., 50/50, 30/70, and 15/85 (x/y represents x% of SKUs account for y% of picking requests). Table 6.2 shows that the cost saving of scenario 2 as compared to scenario 1 increases as C_{a1} and C_r increase, and/or when the picking activities become more skewed.

Table 6.1 Layout features of the optimal solution

		Cost	l*(w_1 +w_2) (ft^2)	(w_1+ w_2)/l	w_1/(w_1+ w_2)
Scenario 1	8000 SKUs	12833149	68672	0.78	0.16
	15000 SKUs	26545355	129504	0.62	0.18
	30000 SKUs	59480913	253344	0.48	0.23
Scenario 2	8000 SKUs	11758377	57344	1.14	0.06
	15000 SKUs	24442367	104960	1.03	0.09
	30000 SKUs	55195459	205568	0.60	0.10

Table 6.2 Cost savings of scenario 2 over scenario 1

		8000 SKUs	15000 SKUs	30000 SKUs
C_{a1}	$\times 1$	8.37%	7.92%	7.20%
	$\times 5$	13.04%	19.56%	9.87%
	$\times 10$	16.68%	20.92%	12.86%
C_r	$\times 1$	8.37%	7.92%	7.20%
	$\times 5$	27.08%	28.72%	29.35%
	$\times 10$	37.59%	40.30%	41.85%
Skewness of R_i	50/50	8.37%	7.92%	7.20%
	30/70	11.65%	12.86%	12.93%
	15/85	17.25%	21.26%	24.70%

The proposed heuristic algorithm is implemented in C and run on a Sun 280R server with 2×900MHz UltraSparc-III CPU and 2GB RAM. The coefficient values of the Nelder-Mead method for reflection, expansion, contraction, and shrinking are set as: $\alpha = 1$, $\gamma = 2$, $\beta = 0.5$, and $\delta = 0.5$, as suggested in Nelder and Mead (1965). The three problems with 8000, 15000, and 30000 SKUs are solved, and for each of them the algorithm is run 30 times with different randomly generated starting points. A solution is called near optimal if each of its dimension variables differs from the optimum by at most 1 (i.e., $\max_{i\in\{l,w_1,w_2\}} | y_i^{heu} - y_i^{opt} | \leq 1$), or locally optimal otherwise. Table 6.3 shows the performance of the proposed algorithm, where the upper part shows the average and maximum computational time for the three problems with different starting simplex and the lower part shows the number of times that the algorithm reaches a near-optimal or locally optimal solution and the corresponding maximum optimality gap. The results show that the algorithm is very efficient and converges quickly in all tested cases. The algorithm can also find a near-optimal solution for most of the randomly generated starting points. There are a few cases in problems with 15000 and 30000 SKUs in which the algorithm ends up with a local optimal solution, but the corresponding optimality gaps are not significant (i.e., within 1.43% for the problem with 15000 SKUs and within 3.56% for the problem with 30000 SKUs).

Table 6.3 Performance of the heuristic algorithm

		8000 SKUs	15000 SKUs	30000 SKUs
Computational Time (s)	Average	24.45	52.62	128.12
	Max	44.56	78.62	189.73
Optimality	Near Opt Solutions	30(30)	22(30)	27(30)
	Max Gap	0.146%	0.002%	0.001%
	Local Opt Solutions	0(30)	8(30)	3(30)
	Max Gap	-	1.43%	3.56%

The robustness of the design solution with regards to possible long-term-forecasting errors in design parameters is also investigated. The result is similar to what is reported in chapter 4 for the problem where all SKUs are assigned to the forward area, and therefore are not reported here in detail. Mostly, the penalty cost due to inexact information at the design stage is within 1% of the optimal cost when the design parameters subject to a forecasting error up to 40%.

6.5 Conclusions

This chapter discusses the general sizing and dimensioning problem in a forward-reserve warehouse that determines the warehouse size and dimensions as well as the forward-reserve allocation to minimize the total warehouse life-cycle cost. The problem is formulated as a mixed integer nonlinear programming model, which is very large for realistic cases. The model is solved with a heuristic algorithm based on the decomposition strategy. Numerical results shows that the heuristic approach is very efficient and can effectively find near-optimal solutions for the cases investigated. It is also shown that cost savings can be achieved by allowing the assignment of SKUs to the forward area based on their flow activities. However, this needs to be balanced against the additional planning and consolidation costs due to picking orders from different storage areas in order to determine the best forward assignment strategy.

CHAPTER 7

CONCLUSIONS AND FUTURE RESEARCH

Warehousing is ubiquitous in logistics and supply chain. A comprehensive literature survey is conducted on warehouse design and operation problems. The results show that previous research had been focused on warehouse operation planning problems, while warehouse design received less attention in the academic research field. However, warehouse design is very important since it provides the framework for on warehouse operations and consequently determines the long-term warehouse life cycle cost.

This research develops an optimization-based approach to designing a forward-reserve warehouse with the objective to minimize its total warehouse life-cycle cost including construction cost, inventory holding and replenishment cost, and material handling cost. The problem is complex due to the various tradeoffs among the different cost elements and the large number of decision variables. A large mixed integer nonlinear optimization model is developed. The solution algorithm uses a decomposition strategy that divides the problem into several smaller problems that are easier to solve and can be coordinated to find the solution of the original problem.

Two decomposition methods are explored: the first is the Generalized Benders Decomposition method, which applies to the special case where all SKUs are assigned to the forward area; the second is a pattern search based method, which applies to the general case that allows the assignment of SKUs to the forward area based on their flow activities. Numerical results demonstrates that the problems can be efficiently solved with the proposed methods, and the resulting optimal (or near optimal) solutions are robust with regards to possible forecasting errors in design parameters that are unavoidable in any design problem.

As far as we know, this research is the first that provides mathematical models and solution algorithms for comprehensive warehouse design. Future research in warehouse design can be pursued in the following directions: (1) Develop analytic and/or simulation models for cost, capacity, and throughput with different warehouse layouts and operational scenarios. These models should provide accurate performance evaluations for different design alternatives and are fundamental to any warehouse design project. Current research in this direction has been mainly focused on order picking cost models for AS/RS. This greatly limits our ability in modeling and solving integrated warehouse design problems, as well as in warehouse performance evaluation. Extending the research to other performance measures (e.g., throughput) for different warehousing systems (e.g., sortation system) is required to improve the practice of warehouse design. (2) This research has been focused on a single but common warehouse type, i.e., the forward-reserve warehouse. Future research should extend the models and solution methods to more general warehouses that have multiple departments and more complex material flows. This depends not only on our capability in providing accurate performance evaluation models as discussed in (1), but also on our capability in integrating these models into a design model which can be efficiently solved to provide a satisfactory solution. The proposed decomposition strategy, especially the pattern search based method, appears a promising approach since it allows combining different sub-models into an integrated design model and solving them in a systematical way to find an optimal or near-optimal solution. (3) Evaluate the robustness of design decisions with regards to specific operational policies, which has important implications on model development and solution method. For example, if the design is robust with regards to operational policies, the design model needs not to represent full operational details, which usually can be much simplified and easier to solve. It is also important since in the early design stage there is a great uncertainty about how the warehouse is going to be operated. If the design is not robust, an optimal design made based on forecasted operational policies might turn out to be a bad solution when the warehouse is actually in operation. Numerical results in this research show that the sizing and dimensioning decision is quite robust with regards to operational policies from the perspective of the total warehouse life cycle cost. This is probable due to the fact that the life cost model includes different cost factors (e.g., construction, inventory, and material handling), which respond differently to any specific change in operational polices and therefore counteracts with each other so that the total life cycle cost is less affected. Future research needs to be conducted to investigate whether this is true for general warehouse design problems.

REFERENCES

Armstrong, R. D., Cook, W. D. and Saipe, A. L. (1979). Optimal batching in a semi-automated order picking system. Journal of the Operational Research Society, 30(8): 711-720.

Ascheuer, N., Grotschel, M. and Abdel-Hamid, A. A.-A. (1999). Order picking in an automatic warehouse: solving online asymmetric TSPs. Mathematical Methods of Operations Research, 49(3): 501-515.

Ashayeri, J., Gelders, L. and Wassenhove, L. V. (1985). A microcomputer-based optimization model for the design of automated warehouses. International Journal of Production Research, 23(4): 825-839.

Ashayeri, J. and Gelders, L. F. (1985). Warehouse design optimization. European Journal of Operational Research, 21: 285-294.

Ashayeri, J., Heuts, R. M., Valkenburg, M. W. T., Veraart, H. C. and Wilhelm, M. R. (2002). A geometrical approach to computing expected cycle times for zone-based storage layouts in AS/RS. International Journal of Production Research, 40(17): 4467-4483.

Azadivar, F. (1986). Maximizing of the throughput of a computerized automated warehousing system under system constraints. International Journal of Production Research, 24(3): 551-566.

Azadivar, F. (1989). Optimum allocation of resources between the random access and rack storage spaces in an automated warehousing system. International Journal of Production Research, 27(1): 119-131.

Bachers, R., Dangelmaier, W. and Warnecke, H. J. (1988). Selection and use of order-picking strategies in a high-bay warehouse. Material Flow, 5: 233-245.

Bartholdi, J. J. and Eisenstein, D. D. (1996). Bucket Brigades: a self-balancing order-picking system for a warehouse, Working Paper. School of Industrial and Systems Engineering, Georgia Institute of Technology.

Bartholdi, J. J. and Gue, K. R. (2000). Reducing labor costs in an LTL crossdocking terminal. Operations Research, 48(6): 823-832.

Bartholdi, J. J. and Hackman, S. T. (2005). Warehouse & Distribution Science. http://www.warehouse-science.com.

Bartholdi, J. J. and Platzman, L. K. (1986). Retrieval strategies for a carousel conveyor. IIE Transactions, 18(2): 166-173.

Bartholdi, J. J. and Platzman, L. K. (1988). Design of efficient bin-numbering schemes for warehouses. Material Flow, 4: 247-254.

Barton, R. R. and Ivey, J. S. (1996). Nelder-Mead simplex modifications for simulation optimization. Management Science, 42(7): 954-973.

Bassan, Y., Roll, Y. and Rosenblatt, M. J. (1980). Internal layout design of a warehouse. AIIE Transactions, 12(4): 317-322.

Bengu, G. (1995). An optimal storage assignment for automated rotating carousels. IIE Transactions, 27: 105-107.

Berry, J. R. (1968). Elements of warehouse layout. International Journal of Production Research, 7(2): 105-121.

Bhaskaran, K. and Malmborg, C. J. (1989). Modelling the service process in a multi-address warehousing system. Applied Mathematical Modelling, 13(7): 386-396.

Bozer, Y. A. (1985). Optimizing throughput performance in designing order picking systems. PhD thesis. Department of Industrial and Systems Engineering. Atlanta, Georgia, Georgia Institute of Technology.

Bozer, Y. A., Quiroz, M. A. and Sharp, G. P. (1988). An evaluation of alternative control strategies and design issues for automated order accumulation and sortation systems. Material Flow, 4: 265-282.

Bozer, Y. A., Schorn, E. C. and Sharp, G. P. (1990). Geometric approaches to solve the Chebyshev traveling salesman problem. IIE Transactions, 22(3): 238-254.

Bozer, Y. A. and Sharp, G. P. (1985). An empirical evaluation of a general purpose automated order accumulation and sortation system used in batch picking. Material Flow, 2(2): 111-131.

Bozer, Y. A. and White, J. A. (1984). Travel-time models for automated storage/retrieval systems. IIE Transactions, 16(4): 329-338.

Bozer, Y. A. and White, J. A. (1990). Design and performance models for end-of-aisle order picking systems. Management Science, 36(7): 852-866.

Bozer, Y. A. and White, J. A. (1996). A generalized design and performance analysis models for end-of-aisle order-picking systems. IIE Transactions, 28: 271-280.

Brooke, A., Kendrick, D., Meeraus, A., Ramesh, R. and Rosenthal, R. E. (1998). GAMS: a user's guide. GAMS Development Corporation.

Brynzer, H. and Johansson, M. I. (1995). Design and performance of kitting and order picking systems. International Journal of Production Economics, 41: 115-125.

Brynzer, H. and Johansson, M. I. (1996). Storage location assignment: using the product structure to reduce order picking times. International Journal of Production Economics, 46-47: 595-603.

Burkard, R. E., Fruhwirth, B. and Rote, G. (1995). Vehicle routing in an automated warehouse: analysis and optimization. Annals of Operations Research, 57: 29-44.

Caron, F., Marchet, G. and Perego, A. (1998). Routing policies and COI-based storage policies in picker-to-part systems. International Journal of Production Research, 36(3): 713-732.

Caron, F., Marchet, G. and Perego, A. (2000). Optimal layout in low-level picker-to-part systems. International Journal of Production Research, 38(1): 101-117.

Chang, D.-T. and Wen, U.-P. (1997). The impact of rack configuration on the speed profile of the storage and retrieval machine. IIE Transactions, 29: 525-531.

Chang, D.-T., Wen, U.-P. and Lin, J. T. (1995). The impact of acceleration/deceleration on travel-time models for automated storage/retrieval systems. IIE Transactions, 27: 108-111.

Chang, S.-H. and Egbelu, P. J. (1997). Relative pre-positioning of storage/retrieval machines in automated storage/retrieval systems to minimize maximum system response time. IIE Transactions, 29: 303-312.

Chew, E. P. and Tang, L. C. (1999). Travel time analysis for general item location assignment in a rectangular warehouse. European Journal of Operational Research, 112: 582-597.

Chow, W.-M. (1986). An analysis of automated storage and retrieval systems in manufacturing assembly lines. IIE Transactions, 18(2): 204-214.

Christofides, N. and Colloff, I. (1972). The rearrangement of items in a warehouse. Operations Research, 21: 577-589.

Cormier, G., Ed. (1987). On the scheduling of order-picking operations in single-aisle automated storage and retrieval systems. Modern Production Management Systems, Elsevier Science Publishers B.V.

Cormier, G. and Gunn, E. A. (1996). On coordinating warehouse sizing, leasing and inventory policy. IIE Transactions, 28: 149-154.

Cormier, G. and Kersey, D. F. (1995). Conceptual design of a warehouse for just-in-time operations in a bakery. Computers and Industrial Engineering, 29(1-4): 361-365.

Cox, B. (1986). Determining economic levels of automation by using a hierarchy of productivity ratios techniques. Proceedings of 7th International Conference on Automation in Warehousing.

Daniels, R. L., Rummel, J. L. and Schantz, R. (1998). A model for warehouse order picking. European Journal of Operational Research, 105: 1-17.

de Koster, M. B. M., van der Poort, E. S. and Wolters, M. (1999). Efficient orderbatching methods in warehouse. International Journal of Production Research, 37(7): 1479-1504.

de Koster, R. (1994). Performance approximation of pick-to-belt orderpicking systems. European Journal of Operational Research, 72(3): 558-573.

de Koster, R. and van der Poort, E. S. (1998). Routing orderpickers in a warehouse: a comparison between optimal and heuristic solutions. IIE Transactions, 30: 469-480.

Eben-Chaime, M. (1992). Operations sequencing in automated warehousing systems. International Journal of Production Research, 30(10): 2401-2409.

Egbelu, P. J. (1991). Framework for dynamic positioning of storage/retrieval machines in an automated storage/retrieval system. International Journal of Production Research, 29(1): 17-37.

Egbelu, P. J. and Wu, C.-T. (1993). A comparison of dwell point rules in an automated storage/retrieval system. International Journal of Production Research, 31(11): 2515-2530.

Elsayed, E. A. (1981). Algorithms for optimal material handling in automatic warehousing systems. International Journal of Production Research, 19(5): 525-535.

Elsayed, E. A. and Lee, M.-K. (1996). Order processing in automated storage/retrieval systems with due dates. IIE Transactions, 28(7): 567-577.

Elsayed, E. A., Lee, M.-K., Kim, S. and Scherer, E. (1993). Sequencing and batching procedures for minimizing earliness and tardiness penalty of order retrievals. International Journal of Production Research, 31(3): 727-738.

Elsayed, E. A. and Stern, R. G. (1983). Computerized algorithms for order processing in automated warehousing systems. International Journal of Production Research, 21(4): 579-586.

Elsayed, E. A. and Unal, O. I. (1989). Order batching algorithms and travel-time estimation for automated storage/retrieval systems. International Journal of Production Research, 27(7): 1097-1114.

Eynan, A. and Rosenblatt, M. J. (1993). An interleaving policy in automated storage/retrieval systems. International Journal of Production Research, 31(1): 1-18.

Eynan, A. and Rosenblatt, M. J. (1994). Establishing zones in single-command class-based rectangular AS/RS. IIE Transactions, 26(1): 38-46.

Foley, R. and Frazelle, E. H. (1991). Analytical results for miniload throughput and the distribution of dual command travel time. IIE Transactions, 23(3): 273-281.

Foley, R., Frazelle, E. H. and Park, B. C. (2002). Throughput bounds for miniload automated storage/retrieval systems. IIE Transactions, 34(10): 915-920.

Francis, R. L. (1967). On some problems of rectangular warehouse design and layout. The Journal of Industrial Engineering, 18: 595-604.

Frazelle, E. H. (2001). World class warehousing and material handling. McGraw-Hill.

Frazelle, E. H., Hackman, S. T., Passy, U. and Platzman, L. K., Eds. (1994). The forward-reserve problem. Optimization in Industry 2. New York, John Wiley & Sons Ltd.

Gademann, A. J. R. M. N., van den Berg, J. P. and van der Hoff, H. H. (2001). An order batching algorithm for wave picking in a parallel-aisle warehouse. IIE Transactions, 33: 385-398.

Gallego, G., Queyranne, M. and Simchi-Levi, D. (1996). Single resource multi-item inventory systems. Operations Research, 44(4): 580-595.

Geofferion, A. M. (1972). Generalized benders decomposition. Journal of Optimization Theory and Applications, 10(4): 237-260.

Ghosh, J. B. and Wells, C. E. (1992). Optimal retrieval strategies for carousel conveyors. Mathematical Computer Modelling, 16(10): 59-70.

Gibson, D. R. and Sharp, G. P. (1992). Order batching procedures. European Journal of Operational Research, 58(1): 57-67.

Goetschalckx, M. (1998). A review of unit load storage policies in warehouse operations. Proceedings of EURO XVI Conference, Brussels, July 12-15.

Goetschalckx, M. and Ratliff, H. D. (1988a). An efficient algorithm to cluster order picking items in a wide aisle. Engineering Costs and Production Economics, 13: 263-271.

Goetschalckx, M. and Ratliff, H. D. (1988b). Order picking in an aisle. IIE Transactions, 20(1): 53-62.

Goetschalckx, M. and Ratliff, H. D. (1988c). Sequencing picking operations in a man-aboard order picking system. Material Flow, 4: 255-263.

Goetschalckx, M. and Ratliff, H. D. (1990). Shared storage policies based on the duration stay of unit loads. Management Science, 36(9): 1120-1132.

Goetschalckx, M. and Ratliff, H. D. (1991). Optimal lane depths for single and multiple products in block stacking storage systems. IIE Transactions, 23(3): 245-258.

Goh, M., Ou, J. and Teo, C.-P. (2001). Warehouse sizing to minimize inventory and storage costs. Naval Research Logistics, 48(4): 299-312.

Graves, S. C., Hausman, W. H. and Schwarz, L. B. (1977). Storage-retrieval interleaving in automatic warehousing systems. Management Science, 23(9): 935-945.

Gray, A. E., Karmarkar, U. S. and Seidmann, A. (1992). Design and operation of an order-consolidation warehouse: models and applications. European Journal of Operational Research, 58: 14-36.

Gudehus, T. (1973). Principles of order picking: operations in distribution and warehousing systems, (in German). Essen, West Germany.

Gue, K. R. (1999). The effects of trailer scheduling on the layout of freight terminals. Transportation Science, 33(4): 419-428.

Guenov, M. and Raeside, R. (1992). Zone shapes in class based storage and multicommand order picking when storage/retrieval machines are used. European Journal of Operational Research, 58: 37-47.

Hackman, S. T., Frazelle, E. H., Griffin, P. M., Griffin, S. O. and Vlasta, D. A. (2001). Benchmarking warehouse and distribution operations: an input-output approach. Journal of Productivity Analysis, 16: 79-100.

Hackman, S. T. and Rosenblatt, M. J. (1990). Allocating items to an automated storage and retrieval system. IIE Transactions, 22(1): 7-14.

Hall, R. W. (1993). Distance approximation for routing manual pickers in a warehouse. IIE Transactions, 25(4): 76-87.

Han, M. H. and McGinnis, L. F. (1986). Carousel application for work-in-process: modelling and analysis. Atlanta, Georgia, Material Handling Research Center, Georgia Institute of Technology.

Han, M. H., McGinnis, L. F., Shieh, J. S. and White, J. A. (1987). On sequencing retrievals in an automated storage/retrieval system. IIE Transactions, 19(1): 56-66.

Han, M. H., McGinnis, L. F. and White, J. A. (1988). Analysis of rotary rack operation. Material Flow, 4: 283-293.

Hariga, M. A. and Jackson, P. L. (1996). The warehouse scheduling problem: formulation and algorithms. IIE Transactions, 28: 115-127.

Harmatuck, D. J. (1976). A comparison of two approaches to stock location. The Logistics and Transportation Review, 12(4): 282-284.

Hausman, W. H., Schwarz, L. B. and Graves, S. C. (1976). Optimal storage assignment in automatic warehousing systems. Management Science, 22(6): 629-638.

Heskett, J. L. (1963). Cube-per-order index - a key to warehouse stock location. Transportation and Distribution Management, 3: 27-31.

Heskett, J. L. (1964). Putting the Cuber-Per-Order Index to work in warehouse layout. Transportation and Distribution Management, 4: 23-30.

Hodgson, T. J. and Lowe, T. J. (1982). Production lot sizing with material-handling cost considerations. IIE Transactions, 14(1): 44-51.

Hung, M. S. and Fisk, C. J. (1984). Economic sizing of warehouses - a linear programming approach. Computers and Operations Research, 11(1): 13-18.

Hwang, H., Baek, W. and Lee, M.-K. (1988). Clustering algorithms for order picking in an automated storage and retrieval system. International Journal of Production Research, 26(2): 189-201.

Hwang, H., Kim, C.-S. and Ko, K.-H. (1999). Performance analysis of carousel systems with double shuttle. Computers and Industrial Engineering, 36: 473-485.

Hwang, H. and Ko, C. S. (1988). A study on multi-aisle system served by a single storage/retrieval machine. International Journal of Production Research, 26(11): 1727-1737.

Hwang, H. and Lee, M.-K. (1988). Order batching algorithms for a man-on-board automated storage and retrieval system. Engineering Costs and Production Economics, 13: 285-294.

Hwang, H. and Lee, S. B. (1990). Travel-time models considering the operating characteristics of the storage and retrieval machine. International Journal of Production Research, 28(10): 1779-1789.

Hwang, H. and Lim, J. M. (1993). Deriving an optimal dwell point of the storage/retrieval machine in an automated storage/retrieval system. International Journal of Production Research, 31(11): 2591-2602.

Hwang, H. and Song, J. Y. (1993). Sequencing picking operations and travel time models for man-on-board storage and retrieval warehousing system. International Journal of Production Economics, 29: 75-88.

Ito, T., Abadi, J. and Mousavi, S. M. (2002). Agent-based material handling and inventory planning in warehouse. Journal of Intelligent Manufacturing, 13(3): 201-210.

Jaikumar, R. and Solomon, M. M. (1990). Dynamic operational policies in an automated warehouse. IIE Transactions, 22(4): 370-376.

Jarvis, J. M. and McDowell, E. D. (1991). Optimal product layout in an order picking warehouse. IIE Transactions, 23(1): 93-102.

Johnson, M. E. (1998). The impact of sorting strategies on automated sortation system performance. IIE Transactions, 30: 67-77.

Johnson, M. E. and Lofgren, T. (1994). Model decomposition speeds distribution center design. Interfaces, 24(5): 95-106.

Kallina, C. and Lynn, J. (1976). Application of the cube-per-order index rule for stock location in a distribution warehouse. Interfaces, 7(1): 37-46.

Karasawa, Y., Nakayama, H. and Dohi, S. (1980). Trade-off analysis for optimal design of automated warehouses. International Journal of Systems Science, 11(5): 567-576.

Keserla, A. and Peters, B. A. (1994). Analysis of dual-shuttle automated storage/retrieval systems. Journal of Manufacturing Systems, 13(6): 424-434.

Kim, B., -I., Graves, R. J., Heragu, S. S. and Onge, A. S. (2002). Intelligent agent modeling of an industrial warehousing problem. IIE Transactions, 34(7): 601-612.

Kim, J. and Seidmann, A. (1990). A framework for the exact evaluation of expected cycle times in automated storage systems with full-turnover item allocation and random service requests. Computers and Industrial Engineering, 18(4): 601-612.

Koh, S. G., Kim, B. S. and Kim, B. N. (2002). Travel time model for the warehousing system with a tower crane S/R machine. Computers and Industrial Engineering, 43(3): 495-507.

Kouvelis, P. and Papanicolaou, V. (1995). Expected travel time and optimal boundary foumulas for a two-class-based automated storage/retrieval system. International Journal of Production Research, 33(10): 2889-2905.

Kulturel, S., Ozdemirel, N. E., Sepil, C. and Bozkurt, Z. (1999). Experimental investigation of shared storage assignment policies in automated storage/retrieval systems. IIE Transactions, 31(8): 739-49.

Lai, K. K., Xue, J. and Zhang, G. (2002). Layout design for a paper reel warehouse: a two-stage heuristic approach. International Journal of Production Economics, 75(3): 231-243.

Larson, N., March, H. and Kusiak, A. (1997). A heuristic approach to warehouse layout with class-based storage. IIE Transactions, 29: 337-348.

Lee, H. S. (1997). Performance analysis for automated storage and retrieval systems. IIE Transactions, 29: 15-28.

Lee, H. S. and Schaefer, S. K. (1996). Retrieval sequencing for unit-load automated storage and retrieval systems with multiple openings. International Journal of Production Research, 34(10): 2943-2962.

Lee, H. S. and Schaefer, S. K. (1997). Sequencing methods for automated storage and retrieval systems with dedicated storage. Computers and Industrial Engineering, 32(2): 351-362.

Lee, M.-K. (1992). A storage assignment policy in a man-on-board automated storage/retrieval system. International Journal of Production Research, 30(10): 2281-2292.

Lee, M.-K. and Kim, S.-Y. (1995). Scheduling of storage/retrieval orders under a just-in-time environment. International Journal of Production Research, 33(12): 3331-3348.

Lee, Y. H., Tanchoco, J. M. A. and Chun, S. J. (1999). Performance estimation models for AS/RS with unequal sized cells. International Journal of Production Research, 37(18): 4197-4216.

Levy, J. (1974). The optimal size of a storage facility. Naval Research Logistics Quarterly, 21(2): 319-326.

Lin, C.-H. and Lu, I.-Y. (1999). The procedure of determining the order picking strategies in distribution center. International Journal of Production Economics, 60-61: 301-307.

Linn, R. J. and Wysk, R. A. (1987). An analysis of control strategies for an automated storage/retrieval system. INFOR, 25(1): 66-83.

Linn, R. J. and Wysk, R. A. (1990). An expert system framework for automated storage and retrieval system control. Computers and Industrial Engineering, 18(1): 37-48.

Linn, R. J. and Xie, X. D. (1993). A simulation analysis of sequencing rules for ASRS in a pull-based assembly facility. International Journal of Production Research, 31(10): 2355-2367.

Lowe, T. J., Francis, R. L. and Reinhardt, E. W. (1979). A greedy network flow algorithm for a warehouse leasing problem. AIIE Transactions, 11(3): 170-182.

Luxhoj, J. T., Agnihotri, D., Kazunas, S. and Nambiar, S. (1993). A prototype knowledge-based system (KBS) for selection of inventory control policies. International Journal of Production Research, 31(7): 1709-1720.

Luxhoj, J. T. and Skarpness, B. O. (1986). A manpower planning model for a distribution center: a case study. Material Flow, 3: 251-261.

Lewis, R.M., and Torczon, V. (1999). Pattern search algorithms for bound constrained minimization. SIAM Journal of Optimization, 9(4): 1082-1099.

Lewis, R.M., and Torczon, V. (2000). Pattern search methods for linearly constrained minimization. SIAM Journal of Optimization, 10(3): 917-941.

Mahajan, S., Rao, B. V. and Peters, B. A. (1998). A retrieval sequencing heuristic for miniload end-of-aisle automated storage/retrieval systems. International Journal of Production Research, 36(6): 1715-1731.

Makris, P. A. and Giakoumakis, I. G. (2003). *k*-Interchange heuristic as an optimization procedure for material handling applications. Applied Mathematical Modelling, 27(5): 345-358.

Mallette, A. J. and Francis, R. L. (1972). A generalized assignment approach to optimal facility layout. AIIE Transactions, 4(2): 144-147.

Malmborg, C. J. (1995). Optimization of cube-per-order index warehouse layouts with zoning constraints. International Journal of Production Research, 33(2): 465-482.

Malmborg, C. J. (1996). An integrated storage system evaluation model. Applied Mathematical Modelling, 20(5): 359-370.

Malmborg, C. J. (2000). Interleaving models for the analysis of twin shuttle automated storage and retrieval systems. International Journal of Production Research, 38(18): 4599-4610.

Malmborg, C. J. (2001). Rule of thumb heuristics for configuring storage racks in automated storage and retrieval systems design. International Journal of Production Research, 39(3): 511-527.

Malmborg, C. J. (2003). Design optimization models for storage and retrieval systems using rail-guided vehicles. Applied Mathematical Modelling, 27(12): 929-941.

Malmborg, C. J. and Al-Tassan, K. (1998). Analysis of storage assignment policies in less than unit load warehousing systems. International Journal of Production Research, 36(12): 3459-3475.

Malmborg, C. J. and Al-Tassan, K. (2000). An integrated performance model for order picking systems with randomized storage. Applied Mathematical Modelling, 24(2): 95-111.

Malmborg, C. J., Balachandran, S. and Kyle, D. M. (1986). A model based evaluation of a commonly used rule of thumb for warehouse layout. Applied Mathematical Modelling, 10(2): 133-138.

Malmborg, C. J. and Deutsch, S. J. (1988). A stock location model for dual address order picking systems. IIE Transactions, 20(1): 44-52.

Malmborg, C. J. and Krishnakumar, B. (1987). On the optimality of the cube per order index for conventional warehouses with dual command cycles. Material Flow, 4: 169-175.

Malmborg, C. J. and Krishnakumar, B. (1989). Optimal storage assignment policies for multiaddress warehousing systems. IEEE Transactions on Systems, Man, and Cybernetics, 19(1): 197-204.

Malmborg, C. J. and Krishnakumar, B. (1990). A revised proof of optimality for the cube-per-order index rule for stored item location. Applied Mathematical Modelling, 14(2): 87-95.

Malmborg, C. J., Krishnakumar, B. and Simons, G. R. (1988). A mathematical overview of warehousing systems with single/dual order-picking cycles. Applied Mathematical Modelling, 12(1): 2-8.

Marsh, W. H. (1979). Elements of block storage design. International Journal of Production Research, 17(4): 377-394.

Marsh, W. H. (1983). A comparison with Berry. International Journal of Production Research, 21(2): 163-172.

Matson, J. O. and White, J. A. (1981). Storage system optimization. Atlanta, Georgia, Production and Distribution Research Center, Georgia Institute of Technology.

McGinnis, L. F. (2003). Best of Breed Warehouse Performance Assessment. Council on Logistics Management Annual Conference, Chicago, IL (see also http://www.isye.gatech.edu/ideas/).

Meller, R. D. (1997). Optimal order-to-lane assignments in an order accumulation/sortation system. IIE Transactions, 29: 293-301.

Meller, R. D. and Mungwattana, A. (1997). Multi-shuttle automated storage/retrieval systems. IIE Transactions, 29(10): 925-938.

Moder, J. J. and Thornton, H. M. (1965). Quantitative analysis of the factors affecting floor space utilization of palletized storage. The Journal of Industrial Engineering, 16(1): 8-18.

Montulet, P., Langevin, A. and Riopel, D. (1998). Minimizing the peak load: an alternate objective for dedicated storage policies. International Journal of Production Research, 36(5): 1369-1385.

Muralidharan, B., Linn, R. J. and Pandit, R. (1995). Shuffling heuristics for the storage location assignment in an AS/RS. International Journal of Production Research, 33(6): 1661-1672.

Nelder, J. A. and Mead, R. (1965). A simplex method for function minimization. Computer Journal, 7: 308-313.

Pan, C.-H. and Liu, S.-Y. (1995). A comparative study of order batching algorithms. Omega International Journal of Management Science, 23(6): 691-700.

Pan, C.-H. and Wang, C.-H. (1996). A framework for the dual command cycle travel time model in automated warehousing systems. International Journal of Production Research, 34(8): 2099-2117.

Pandit, R. and Palekar, U. S. (1993). Response time considerations for optimal warehouse layout design. Journal of Engineering for Industry, 115: 322-328.

Park, B. C., Foley, R. D., White, J. A. and Frazelle, E. H. (2003). Dual command travel times and miniload system throughput with turnover-based storage. IIE Transactions, 35(4): 343-355.

Park, B. C., Frazelle, E. H. and White, J. A. (1999). Buffer sizing models for end-of-aisle order picking systems. IIE Transactions, 31: 31-38.

Park, Y. H. and Webster, D. B. (1989). Modelling of three-dimensional warehouse systems. International Journal of Production Research, 27(6): 985-1003.

Perlmann, A. M. and Bailey, M. (1988). Warehouse logistics systems - a CAD model. Engineering Costs and Production Economics, 13: 229-237.

Peters, B. A., Smith, J. S. and Hale, T. S. (1996). Closed form models for determining the optimal dwell point location in automated storage and retrieval systems. International Journal of Production Research, 34(6): 1757-1771.

Petersen, C. G. (1997). An evaluation of order picking routing policies. International Journal of Operations and Management Science, 17(11): 1098-1111.

Petersen, C. G. (1999). The impact of routing and storage policies on warehouse efficiency. International Journal of Operations and Production Management, 19(10): 1053-1064.

Petersen, C. G. (2000). An evaluation of order picking policies for mail order companies. Production and Operations Management, 9(4): 319-335.

Petersen, C. G. (2002). Considerations in order picking zone configuration. International Journal of Operations and Production Management, 22(7): 793-805.

Pliskin, J. S. and Dori, D. (1982). Ranking alternative warehouse area assignments: a multiattribute approach. IIE Transactions, 14(1): 19-26.

Randhawa, S. U., McDowell, E. D. and Wang, W.-T. (1991). Evaluation of scheduling rules for single- and dual-dock automated storage/retrieval system. Computers and Industrial Engineering, 20(4): 401-410.

Randhawa, S. U. and Shroff, R. (1995). Simulation-based design evaluation of unit load automated storage/retrieval systems. Computers and Industrial Engineering, 28(1): 71-79.

Rao, A. K. and Rao, M. R. (1998). Solution procedures for sizing of warehouses. European Journal of Operational Research, 108: 16-25.

Ratliff, H. D. and Rosenthal, A. S. (1983). Order-picking in a rectangular warehouse: a solvable case of the traveling salesman problem. Operations Research, 31(3): 507-521.

Roberts, S. D. and Reed, R. (1972). Optimal warehouse bay configurations. AIIE Transactions, 4(3): 178-185.

Roll, Y. and Rosenblatt, M. J. (1987). Shifting in warehouses. Material Flow, 4: 147-157.

Roll, Y., Rosenblatt, M. J. and Kadosh, D. (1989). Determining the size of a warehouse container. International Journal of Production Research, 27(10): 1693-1704.

Roodbergen, K. J. and de Koster, R. (2001a). Routing methods for warehouses with multiple cross aisles. International Journal of Production Research, 39(9): 1865-1883.

Roodbergen, K. J. and de Koster, R. (2001b). Routing order pickers in a warehouse with a middle aisle. European Journal of Operational Research, 133: 32-43.

Rosenblatt, M. J. and Eynan, A. (1989). Deriving the optimal boundaries for class-based automatic storage/retrieval systems. Management Science, 35(12): 1519-1524.

Rosenblatt, M. J. and Roll, Y. (1984). Warehouse design with storage policy considerations. International Journal of Production Research, 22(5): 809-821.

Rosenblatt, M. J. and Roll, Y. (1988). Warehouse capacity in a stochastic environment. International Journal of Production Research, 26(12): 1847-1851.

Rosenblatt, M. J., Roll, Y. and Zyser, V. (1993). A combined optimization and simulation approach for designing automated storage/retrieval systems. IIE Transactions, 25(1): 40-50.

Rosenwein, M. B. (1994). An application of cluster analysis to the problem of locating items within a warehouse. IIE Transactions, 26(1): 101-103.

Rosenwein, M. B. (1996). A Comparison of Heuristics for the Problem of Batching Orders for Warehouse Selection. International Journal of Production Research, 34(3): 657-664.

Ross, A. and Droge, C. (2002). An integrated benchmarking approach to distribution center performance using DEA modeling. Journal of Operations Management, 20: 19-32.

Rowenhorst, B., Reuter, B., Stockrahm, V., van Houtum, G. J., Mantel, R. J. and Zijm, W. H. M. (2000). Warehouse design and control: framework and literature review. European Journal of Operational Research, 122: 515-533.

Ruben, R. A. and Jacobs, F. R. (1999). Batch Construction Heuristics and Storage Assignment Strategies for Walk/Ride and Pick Systems. Management Science, 45(4): 575-596.

Ryoo, H. S. and Sahinidis, N. V. (1996). A branch-and-reduce approach to global optimization. Journal of Global Optimization, 8: 107-138.

Sadiq, M., Landers, T. L. and Taylor, G. D. (1996). An assignment algorithm for dynamic picking systems. IIE Transactions, 28: 607-616.

Sarker, B. R., Mann, L. and Santos, J. D. (1994). Evaluation of a class-based storage scheduling technique applied to dual-shuttle automated storage and retrieval systems. Production Planning & Control, 5(5): 442-449.

Sarker, B. R., Sabapathy, A., Lal, A. M. and Han, M. (1991). The performance evaluation of a double shuttle automated storage retrieval system. Production Planning & Control, 2(3): 207-213.

Schefczyk, M. (1993). Industrial benchmarking: a case study of performance analysis techniques. International Journal of Production Economics, 32: 1-11.

Schwarz, L. B., Graves, S. C. and Hausman, W. H. (1978). Scheduling policies for automatic warehousing systems: simulation results. AIIE Transactions, 10(3): 260-270.

Seidmann, A. (1988). Intelligent control schemes for automated storage and retrieval systems. International Journal of Production Research, 26(5): 931-952.

Sharp, G. P. (2000). Warehouse Management, in: Salvendy G. Ed., Handbook of Industrial Engineering, John Wiley & Sons, Inc., New York,

Sharp, G. P., Vlasta, D. A. and Houmas, C. G. (1994). Economics of storage/retrieval systems for item picking. Atlanta, Georgia, Material Handling Research Center, Georgia Institute of Technology.

Thonemann, U. W. and Brandeau, M. L. (1998). Optimal storage assignment policies for automated storage and retrieval systems with stochastic demands. Management Science, 44(1): 142-148.

Torczon, V. (1997). On the convergence of pattern search algorithms. SIAM Journal of Optimization, 7(1): 1-25.

Tsui, L. Y. and Chang, C. H. (1990). A microcomputer based decision support tool for assigning dock doors in freight yards. Computers and Industrial Engineering, 19(1-4): 309-312.

Tsui, L. Y. and Chang, C. H. (1992). An optimal solution to a dock door assignment problem. Computers and Industrial Engineering, 23(1-4): 283-286.

van den Berg, J. P. (1996). Multiple order pick sequencing in a carousel system: a solvable case of the rural postman problem. Journal of Operational Research Society, 47: 1504-1515.

van den Berg, J. P. (2002). Analytic expressions for the optimal dwell point in an automated storage/retrieval system. International Journal of Production Economics, 76(1): 13-25.

van den Berg, J. P. and Gademann, A. J. R. M. N. (1999). Optimal routing in an automated storage/retrieval system with dedicated storage. IIE Transactions, 31: 407-415.

van den Berg, J. P. and Gademann, A. J. R. M. N. (2000). Simulation study of an automated storage/retrieval system. International Journal of Production Research, 38(6): 1339-1356.

van den Berg, J. P., Sharp, G. P., Gademann, A. J. R. M. N. and Pochet, Y. (1998). Forward-reserve allocation in a warehouse with unit-load replenishments. European Journal of Operational Research, 111: 98-113.

van Oudheusden, D. L., Tzen, Y.-J. and Ko, H.-T. (1988). Improving storage and order picking in a person-on-board AS/R system: a case study. Engineering Costs and Production Economics, 13: 273-283.

van Oudheusden, D. L. and Zhu, W. (1992). Storage layout of AS/RS racks based on recurrent orders. European Journal of Operational Research, 58: 48-56.

Vaughan, T. S. and Petersen, C. G. (1999). The effect of warehouse cross aisle on order picking efficiency. International Journal of Production Research, 37(4): 881-897.

Vickson, R. G. (1996). Optimal storage locations in a carousel storage and retrieval system. Location Science, 4(4): 237-245.

Vidal, C. J. and Goetschalckx, M. (1997). Strategic production-distribution models: a critical review with emphasis on global supply chain models. European Journal of Operational Research, 98: 1-18.

Wang, J.-Y. and Yih, Y. (1997). Using neural networks to select a control strategy for automated storage and retrieval systems (AS/RS). International Journal of Computer Integrated Manufacturing, 10(6): 487-495.

Wen, U.-P. and Chang, D.-T. (1988). Picking rules for a carousel conveyor in an automated warehouse. Omega International Journal of Management Science, 16(2): 145-151.

White, J. A., DeMars, N. A. and Matson, J. O. (1981). Optimizing storage system selection. Proceedings of the 4th. International Conference on Automation in Warehousing, Tokyo, Japan.

White, J. A. and Francis, R. L. (1971). Normative models for some warehouse sizing problems. AIIE Transactions, 9(3): 185-190.

Wilson, H. G. (1977). Order quantity, product popularity, and the location of stock in a warehouse. AIIE Transactions, 9(3): 230-237.

Yoon, C. S. and Sharp, G. P. (1995). Example application of the cognitive design procedure for an order pick system: case study. European Journal of Operational Research, 87: 223-246.

Yoon, C. S. and Sharp, G. P. (1996). A structured procedure for analysis and design of order pick systems. IIE Transactions, 28: 379-389.

Zeng, A. Z., Mahan, M. and Fluet, N. (2002). Designing an efficient warehouse layout to facilitate the order-filling process: an industrial distributor's experience. Production and Inventory Management Journal, 43(3-4): 83-88.

Zhang, G., Xue, J. and Lai, K. K. (2000). A genetic algorithm based heuristic for adjacent paper-reel layout problem. International Journal of Production Research, 38(14): 3343-3356.

Zhang, G., Xue, J. and Lai, K. K. (2002). A class of genetic algorithms for multiple-level warehouse layout problems. International Journal of Production Research, 40(3): 731-744.

Zollinger, H. A. (1996). Expanded methodology to concept horizontal transportation problem solutions. Progress in Material Handling Research. Graves, R. J., McGinnis, L. F., Medeiros, D. J., Ward, R. E. and Wilhelm, M. R.**:** 651-663.